



LabSmith HVS448 High Voltage Sequencer

User Guide

Version 1.4

For operation and control of LabSmith HVS448™ High Voltage Sequencer and
Sequence™ Driver Software



This manual © 2011 LabSmith. Sequence Software ©2002-2011 LabSmith, Inc.

No part of this document may be reproduced or distributed without the consent of LabSmith.

LabSmith, HVS448 and Sequence are trademarks of LabSmith, Inc.

Microsoft, Windows NT, XP and 7 are registered trademarks of Microsoft Corporation.

Adobe and Acrobat are registered trademarks of Adobe, Inc.

LabView is a trademark of National Instruments, Inc.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION	7
How You Will Use the HVS448	8
Equipment List	9
Cautions and Warnings	9
Getting Help	10
About This Manual	10
CHAPTER 2. GETTING STARTED	11
Installing Sequence Software	11
Grounding the HVS448 Channels	11
Setting Up the HVS448 Hardware for Operation	12
Using the Sequence Wizards to Create a Sequence File	13
Simple Sequence Wizard	13
Complex Sequence Wizard	15
Running a Sequence	16
CHAPTER 3. HVS448 HARDWARE	19
Front Panel Controls	19
Front Panel LEDs	20
Locking the Front Panel Controls	20
Back Panel Connections	21
Master Interlock Input	21
RS-232 Serial Port Connector	21
High Voltage Connections	21
Digital Connections	22
Disable/Enable HV	22
Recalibrating the Channels	22
CHAPTER 4. SEQUENCE SOFTWARE OVERVIEW	25
Field Upgradeable Firmware	25
Sequence Files	25
Online and Offline Operation	26
Sequence Software Environment	26
Monitor Settings	27
CHAPTER 5. HIGH VOLTAGE POWER SUPPLY: MANUAL MODE	29
Channel Control Panels	29
Channel Settings	30

Channel Status	30
Channel Output Limits	30
Error Reporting/Trip Settings	31
Manual Power Supply Control Panel	32
Overall Settings	33
Monitoring	33
Channel Settings	34
Error Flags	34
View Channel Settings in the Left Pane	35
Using Trace Measurements to Record Data	35
CHAPTER 6. HIGH VOLTAGE SEQUENCING: AUTOMATED MODE	37
HVS448 System Architecture	37
Master and Channel Sequencers	39
Manual Changes During an Automated Sequence	39
Inter-Channel Communication	39
Step Instructions	40
Orderly Step Flow	40
Step H: The Exit Step	40
Flags	41
Triggers	41
Sequence and Program Flow	43
Monitoring and Interacting with an Automated Sequence	43
Sequencer Control Panel	43
Channel Sequence Control Panels	45
The Programming Interface	47
Editing Sequences	48
View and Edit Instructions in the Left Pane	48
View and Edit Instructions in the Right Pane	49
Edit with the Programming Keypad	50
Error Checking and Parsing	51
CHAPTER 7. CHANNEL PROGRAM INSTRUCTIONS	53
Nomenclature	53
Execution Controls	53
Inter-channel Signaling and Triggering	57
Channel Functions	60
Register Functions	61
Calculator Functions	62
Reverse Polish Notation and the Numerical Stack	62
Integral Representation of Real Numbers	63
Conditional Tests	67

APPENDIX 1. MENUS AND TOOLBARS	69
Menu Functions	69
Toolbars	70
APPENDIX 2. PROGRAMMING EXAMPLE	71
Setting up the Experiment	72
Writing the Sequence	72
Running the Sequence	74
APPENDIX 3. TROUBLE SHOOTING – COMMON OPERATIONAL ERRORS	77

CHAPTER 1. INTRODUCTION

The HVS448 High Voltage Sequencer (*Figure 1*) is a unique device that provides eight coordinated channels of high voltage sourcing and sensing, with a powerful sequencing environment that lets you quickly automate complex experiments. If you are working in MEMS, microfluidics or other research, the HVS448 will help you take control of your experiments, so you can concentrate on science.



Figure 1. LabSmith HVS448 High Voltage Sequencer.

The HVS448 provides eight **channels** of high voltage sourcing and sensing. You can set each channel independently (manually or automatically) to supply voltage, supply current, or float at high input resistance, all while continuously monitoring the voltage and current. The maximum supplied voltage, current, measurement resolution and input resistance depends on the model (*Table 1*).

The included Sequence™ software lets you automatically set the functions and parameters of each channel. The set of automation commands, called a **sequence**, is constructed from smaller **programs** that control particular events. Each channel runs its own independent sequence, and channels can also communicate state changes to each other, and respond to external events.

The settings and programming for a particular experiment can be stored as a **sequence file** on your computer.

Table 1. LabSmith HVS448 High Voltage Sequence Available Models.

Model	Max Output Voltage [+V]	Max Voltage Differential [V]	Max Output Current [+mA]	Max Output Current Per Channel [+mA]	Current Monitor Resolution [nA]	Voltage Monitor Resolution [mV]
6000D	3000	6000	3.2	2.4	300	100
6000D-LC	3000	6000	0.16	0.12	15	100
3000	3000	3000	6.0	4.8	300	100
3000-LC	3000	3000	0.3	0.24	15	100
3000D	1500	3000	6.0	4.8	500	50
3000D-LC	1500	3000	0.3	0.24	25	50
1500	1500	1500	12	10	500	50
1500-LC	1500	1500	0.6	0.50	25	50
800	800	800	25	20	1000	25
800-LC	800	800	1.3	1.0	50	25
400	400	400	50	40	2000	12
400-LC	400	400	2.5	2.0	100	12
200	200	200	100	80	4000	6
200-LC	200	200	5.0	4.0	200	6

How You Will Use the HVS448

In **manual mode**, you can set the eight channels of the HVS448 independently to supply voltage, supply current, or float at high input resistance, all while continuously monitoring the voltage and current. You can manually set and monitor functions using the Sequence software interface or LabView®.

In **automated mode**, you'll use the HVS448 as follows:

- In Sequence, program the sequence steps and settings for each channel.
- Store those properties and logic to the HVS448's memory.
- Test your sequence by manually issuing control commands and examining the channel states, using the software, the HVS448's channel LEDs and/or a multimeter.
- Run the experiment, using the HVS448 by itself or with the computer (see [Sequencer Control Panel](#)).

Equipment List

Your HVS448 package should include the following items:

- The HVS448 sequencer
- AC power cable
- Sequence™ software installation CD*
- 9-pin RS232 cable
- 50-Ω Terminator
- HVS448 User's Manual.

You may have also ordered and received these recommended, but not required, items:

- Set of 8 HVS High Voltage Cables
- Set of 8 microclip connectors
- Platinum electrodes
- RS232-to-USB adapter, which allows you to operate the HVS448 from a laptop.

If any of these items are missing or damaged please contact LabSmith support.

(support@labsmith.com.)

Cautions and Warnings

LabSmith, Inc. believes that the HVS448 product information enclosed is an accurate description of the typical characteristics and uses of the product, but it is your responsibility to thoroughly test the product in your specific application to determine its performance, efficacy and safety in your conditions of use.

Standard lab safety rules should always be applied: keep the unit away from liquids and excessive heat, leave space around the unit for heat dissipation, etc.

Understand the characteristics of the inputs and outputs of the HVS448 before connecting other equipment. Failure to do so may result in damage to the HVS448 or to your external devices.



WARNING

The HVS448 is a high voltage power supply and thus presents a serious risk of personal injury if not used in accordance with design and/or use specifications, if it is used in applications on products for it is not intended or designed, or if it is used by untrained or unqualified personnel. Before testing, incorporating or using the HVS448, understand and adhere to all applicable instructions, protocol, cautions and safety procedures.

* LabView drivers and Software Developer's Kit upon request – support@labsmith.com

Getting Help

This guide is your main source for information on operating the HVS448 and Sequence software. Choose **Help>About Sequence**, then click the **Open User Manual** link to browse this manual electronically, in Adobe® Acrobat® PDF format.

The LabSmith web site (www.labsmith.com) also includes helpful applications notes and other technical information to help you use the HVS448. If you are unable to find the information you need, please contact our technical support team at (925) 292-5161, or email us at support@labsmith.com. Be prepared to give the HVS448 serial number (located on the bottom of the unit), software version and firmware version you are using. To determine the software version number, select **Help>About Sequence**. To verify the firmware version, go to **Action>Get Versions**. The firmware version number will be displayed in the status bar. The unit must be Online for the firmware version to be displayed.

About This Manual

Several conventions are used throughout this manual to highlight important features and concepts:

Menu>Option Menu bar selections are shown in bold text, with descending options separated by the “>” symbol. For example, **Help > About Sequence** indicates that you should click the **Help** menu, then click **About Sequence**.

Italics When browsing the electronic version of this manual, click on these blue italic cross-references to view other pertinent sections of the manual.

Note Notes indicate helpful tips or additional information.

CAUTION These notes will help you to avoid losing data or damaging hardware.

 **WARNING** These important notes indicate situations that could result in injury or serious equipment damage. Read and understand all warnings before operating the HVS448.

If you find any errors or omissions in this manual please let us know at support@labsmith.com.

CHAPTER 2. GETTING STARTED

Installing Sequence Software

Sequence software runs on any PC-compatible computer running Microsoft Windows® XP, NT or 7. To install Sequence software:

1. Insert the **Sequence for Windows** CD.
2. Double-click **SETUP.EXE** and follow the on-screen instructions.

To run the software, double-click the **Sequence** icon on your desktop, or choose **LabSmith>Sequence>Sequence** from the **Start** menu.

Grounding the HVS448 Channels

The chassis and the grounds of all the channels are electrically connected to the ground pin of the HVS power connector. Only connect the HVS to a properly grounded outlet.

You have several options for properly grounding the HVS:

- Connect a wire between the grounding pin of on the back of the HVS and other apparatus (e.g., microscopes, etc.) to ensure that all equipment remains at safe voltages despite experimental errors or problems.
- Connect the HVS to ground using one channel or the grounding pin, then use another channel to ground another instrument. This method ensures that the HVS high-voltage signal does not cause voltage across another instrument.
- Attach the grounding pin to a grounding cable to ground all channels.
- Using LabSmith's HVC cables the HVS448 can be grounded by attaching any one of the grounding leads to a ground source (download the [HVC Cable Safety Sheet](#)).



WARNING

Do not handle cable terminations or clips while they are connected to an energized circuit. The micro- and nano-clips do not protect from high voltage. The clips are not designed to be handled when high voltage is present. When using the clips, use the same safety precautions as you would with an exposed electrode.

Cable ends and clips should be housed in an interlocked enclosure to protect the user from accidental exposure to the high voltage circuit (**Figure 2**).

High voltage cables and clips present a serious risk of personal injury if not used in accordance with design and/or use specifications, if they are used in applications for products for which they are not intended or designed, or if they are used by untrained or unqualified personnel. Before testing, incorporating or using the cables with the HVS448,

understand and adhere to all applicable instructions, protocol, cautions and safety procedures.

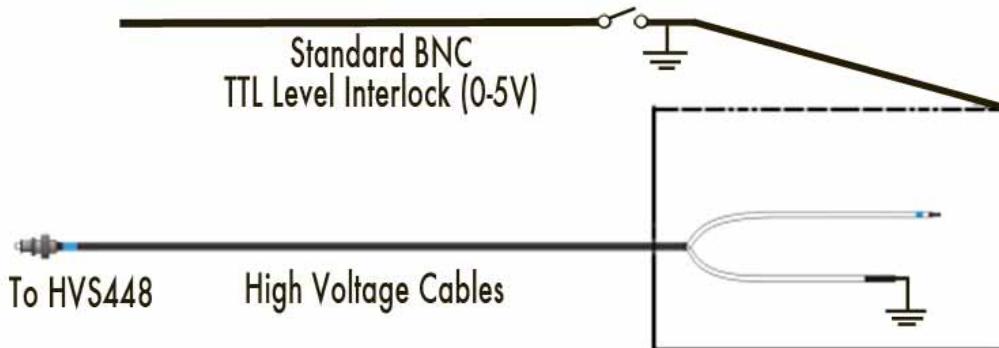


Figure 2. Diagram of LabSmith HVC High Voltage Cables, Shown Grounded and in Interlocked Chamber for Safe Operation.

Note: Grounding any channel or the chassis grounds all of the channels. Using the grounding cable attached to the channel you are applying voltage to may give you less noise than grounding the chassis or a different channel. If noise is a concern, you should use a grounding wire at each channel to which you are applying high voltage.

Setting Up the HVS448 Hardware for Operation

1. Ground your HVS448 using any of the methods recommended above.
2. Connect the $50\text{ M}\Omega$ terminator into the master interlock (see [**Master Interlock Input**](#)).
3. Plug the power cable from the back of the HVS448 into a 90–250 VAC power outlet.
4. Connect the RS232 cable to the back of the HVS448 and to a free serial port on the PC.
5. Plug the power cable from the back of the HVS448 into a 90–250 VAC power outlet. Connect the RS232 cable to the back of the HVS448 and to a free serial port on the PC.

Note: If connecting to a laptop, connect the RS232 cable to the RS232(male)-USB adapter. Plug the USB connector into the laptop's USB port.

6. Press the **Power** button on the front of the HVS448.
7. From your computer, run the Sequence software.
8. From within Sequence, select **View>Options>Communications**. The Serial Communications dialog box will open ([**Figure 3**](#)).

9. Select the COM port to which the HVS448 is connected and click **OK**.

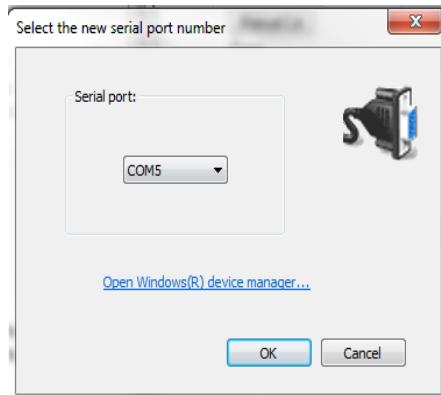


Figure 3. Serial Communications Dialog Box.

Using the Sequence Wizards to Create a Sequence File

Sequence Wizards simplify the process of creating sequences by letting you enter voltages and currents directly for each step of the sequence. A sequence can contain up to 20 steps, with settings for up to 8 channels. These settings, and all the programming instructions necessary to automatically set up each channel, are stored in a Sequence file with a .seq extension.

Simple Sequence Wizard

The Simple Sequence Wizard lets you apply/measure DC current and voltage. To use the Simple Sequence Wizard:

1. Open the Sequence software if it is not already opened.
2. Select **Tools > Simple sequence wizard**. The Wizard interfaces contain tabs for each step (*Figure 4*). When you launch a wizard the settings for Step A will be displayed. Click on any step's tab to view that step's settings.
3. In the **Step Name** field, enter a name which easily identifies what is happening during the step. The name must begin with a capital letter and contain only letters, numbers, “.” or “_”.

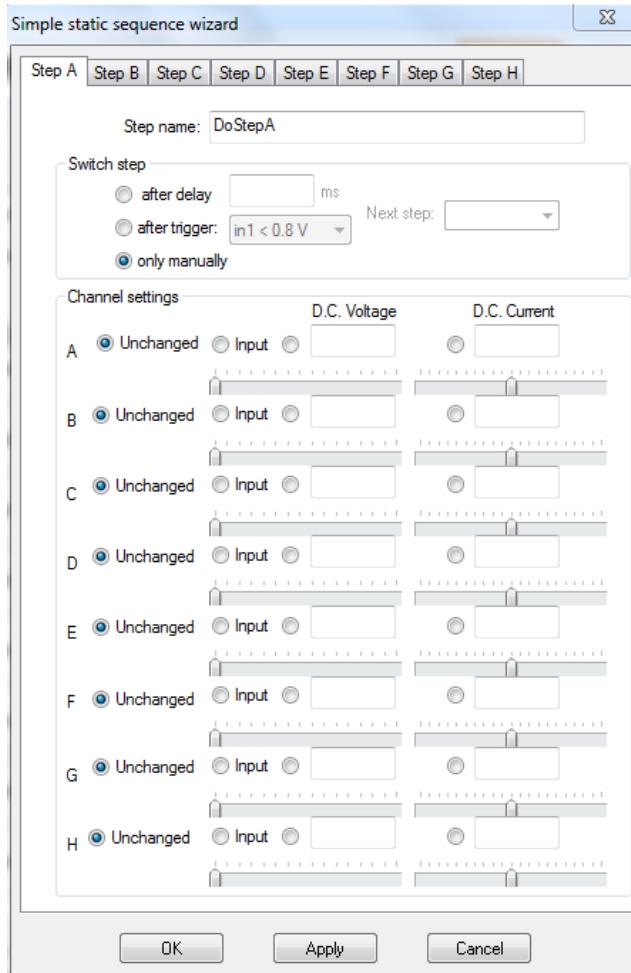


Figure 4. Simple Sequence Wizard Window.

4. Set the **Switch step** settings to specify the condition at which the sequence will move from the current step to the next step in the sequence:
 - a. Select **Only manually** to switch steps manually, using the Master Controller (see *Sequencer Control Panel*).
 - b. Select **After delay**, enter a time period (in milliseconds), and select the **Next Step** to which the sequence should jump after that specified time.
 - c. Select **After trigger**, choose the condition that will trigger a step change, then select the **Next Step** to which the sequence should jump when the trigger condition occurs. The trigger conditions can be a high or low TTL signal at any of the four inputs In1 – In4.
5. Enter the setting changes for each channel during this step:
 - a. Select **Unchanged** to leave the channel in the same state as the previous sequence step.

- b. Select **Input** to use the channel as a voltmeter for an incoming signal.
- c. Select **D.C. Voltage**, then specify the voltage for the channel to maintain during this step. You can either enter the voltage in the box or drag the slider to set the voltage. If you are running in constant current mode, you do not need to specify this value.
- d. Select **D.C. Current**, then specify the current for the channel to maintain during this step. You can either enter the current in the box or drag the slider to set the current. If you are running in constant voltage mode, you do not need to specify this value.

6. Complete 1-4 for each of the steps in the sequence.
7. Click **Apply** to send the program to the HVS448.
8. Choose **File>Save**, then specify a location and file name to which to save the sequence file. It will be saved as a “.seq” file.

Note: Once a file name has been specified you can click **OK** to send the program to the HVS448 and automatically save it as well.

Complex Sequence Wizard

The Complex Sequence Wizard (*Figure 5*) provides the same Name, Next Step and DC operations as described in the Simple wizard above. It also gives you the ability to apply a low frequency (1kHz), sine, cosine, triangular or sawtooth wave to the applied voltage.

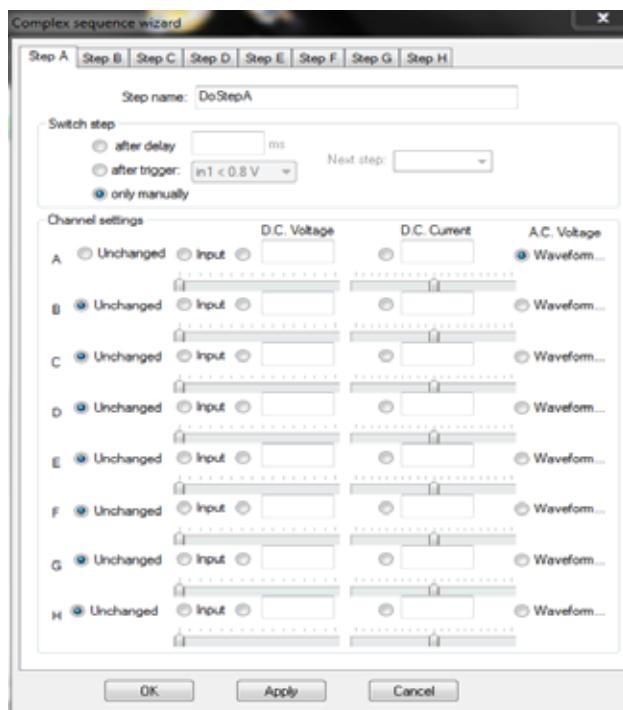


Figure 5. Complex Sequence Wizard.

For AC operations, click the radio button labeled **Waveform** next to each channel to open the **Waveform Generation** dialog box ([Figure 6](#)).

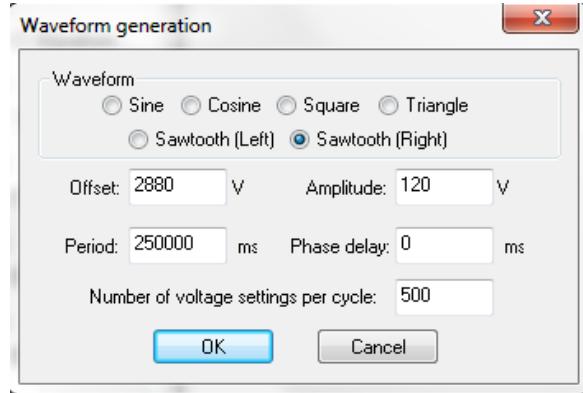


Figure 6. Complex Sequence Wizard: Waveform Generation dialog box.

Choose the **Waveform** shape you wish to apply, then set:

- **Offset (V)**, the DC offset (if any) applied to the waveform.
- **Amplitude (V)**, the peak to peak voltage differential of the waveform.

Note: By default, generated waveforms are bipolar (i.e., they oscillate between negative and positive voltages). When a unipolar waveform is required (for example, a 0 to 5V square wave) the offset and amplitude controls together can be used to adjust the output accordingly.

- **Period (ms)**, the duration of a single cycle of the waveform. The maximum frequency that can be applied is 1 kHz.
- **Phase Delay (ms)**, the delay after a sequence step starts until the waveform begins, and
- **Number of voltage settings per cycle**, the number of times per cycle that the voltage will be updated.

Running a Sequence

To run the saved sequence:

1. Select the sequence file ([File>Open](#)) if it is not already selected.
2. Click the **Online/Offline** toolbar button  to toggle the software online (see [Online and Offline Operation](#)).
 1. Click the **Enable** toolbar button  to enable voltage (see [Disable/Enable HV](#)).
 2. Send the settings to the HVS448.

- a. If the HVS448 is brought online after the program is written and saved, you will see the **HVS is Online** dialog box ([Figure 7](#)). Check the **Send the current sequence program to the HVS448 now** box.
- b. If the HVS448 was online prior to programming choose **Actions>Program All**.

3. In the left pane click **Sequence** to open the *Sequencer Control Panel*.
4. Press **A** at the bottom of the Sequencer Control Panel to run Step A, etc. If you selected timed events or triggers, then the automated sequence you created will take over. If not, and you selected “manual” to switch between steps in the **Sequence Wizard**, click the  **Sequencer** icon in the left pane. The **Manual Sequence Monitor and Control** ([Figure 8](#)) will open in the right pane.
5. Switch between Steps A-H manually by using the cursor to select a step and left clicking on the mouse.

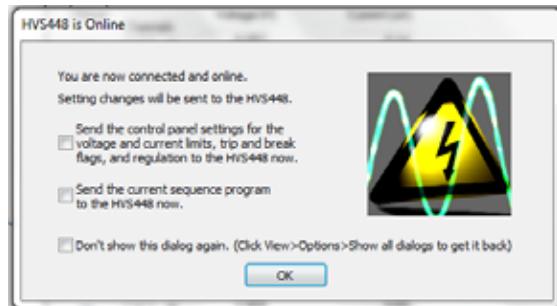


Figure 7. HVS448 is Online Dialog Box.

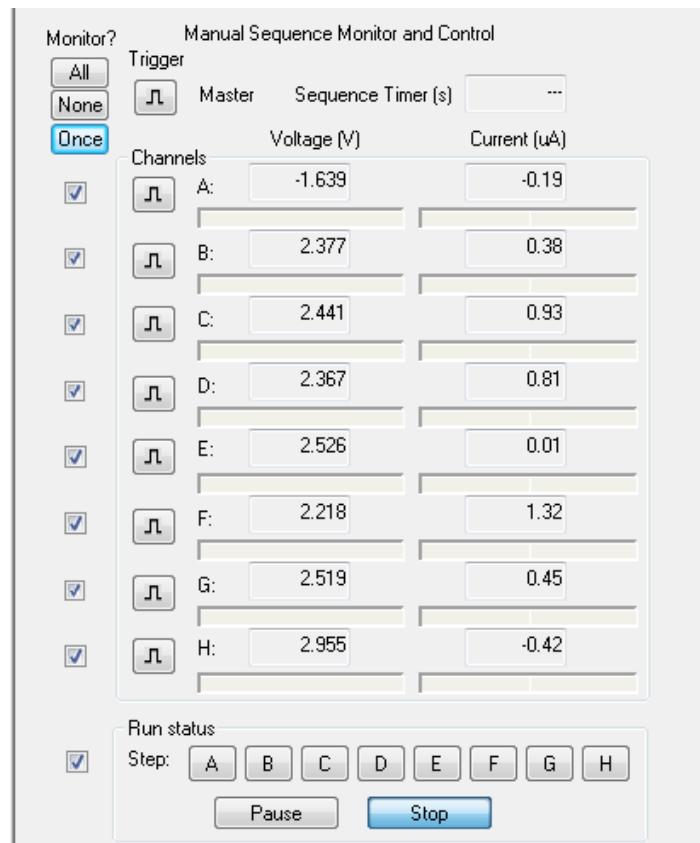


Figure 8. Manual Sequence Monitor and Control window.

CHAPTER 3. HVS448 HARDWARE

Most control of the HVS448 sequencer occurs through the Sequence software environment. The HVS front panel ([Figure 9](#)) also includes several important functions that let you use the unit without a computer.



Figure 9. HVS448 Front Panel (3000V model shown).

Front Panel Controls

The **Power button** and **Power LED** are located in the upper left corner of the front panel. When you first turn on the HVS448, the LEDs will flash green and red while the system runs its initial diagnostics.

While the programming of the HVS448 must be done on a computer, once a sequence file has been sent to the HVS448, it can be operated using the front panel buttons. The front panel buttons typically work in pairs, by either of two methods:

1. Select a channel or step (A–H) and then select a function to perform on that channel/step. When you select the channel/step, its LED will remain lit until you select a function. Push the channel/step button again to deselect it.
2. Select a function, then select the channel(s) to apply that function to. The function button will remain lit after you select a channel, indicating that you can select additional channels. Push the function button again to deselect it. Selecting a different function will also deselect the first function.

The front panel controls are as follows:

A–H	Select individual channels or sequence Steps, A–H.
All	Select all channels/steps, so you can apply a function to all of them at once.
Enable/Disable	Enable/disable the channel(s). The high voltage connectors on the back panel will float to the driving voltage when disabled, regardless of the sequence or the states of other channels.
Trigger	Force the trigger of a channel. The trigger only affects an Await Trigger function in that channel's sequence. (see Inter-Channel Communication).

Push **Trigger**, then **ALL**, to trigger the Master sequencer, which has authority over the individual channel sequencers.

Pause	Pause the current sequence. All channels will remain frozen in their last state. Press the button again to resume the sequence. You do <i>not</i> need to select a channel or step before pushing Pause.
Run	Jump to the selected Step, A–H, in the sequence.
Stop	Stop the sequence if one is running and set all high voltage channels to be inputs (i.e., they neither source nor sink current). You do <i>not</i> need to select a channel or step before pushing Stop.

Front Panel LEDs

LEDs indicate the state of each channel. A red light indicates that the channel is sourcing current (current is flowing out of the channel). A green light indicates that the channel is sinking current (current is flowing into the channel). When a channel LED is off, the channel is an input (not sinking or sourcing current). You can optionally override each channel's LED settings while a sequence is executing, for example, to indicate the execution status (see [LED](#)).

The **ALL LED** function indicates the status of the high voltage interlock. The LED glows green when high voltage is enabled and red when high voltage is disabled, because of an open interlock or over-current, as examples. The LED is off when high voltage has been manually disabled. The LED glows yellow when the HVS448 is switching the high voltage on or off or changing the voltage ranges. During the execution of a sequence, the master sequencer can be optionally programmed to override the ALL LED setting.

Locking the Front Panel Controls

Lock Front Panel prevents accidental or purposeful adjustment of sequences or voltages from the front panel. When the panel is locked, the LED next to the power button glows a steady red and all buttons are dark.

Unlock Front Panel allows front panel control of the HVS448. This is the default condition. When the panel is unlocked, the power LED glows green and the front panel buttons are illuminated.

The Lock functions are nonvolatile: the panel will remain locked or unlocked even if power is removed and restored. The panel can only be unlocked from within the Sequence software.

Back Panel Connections

Master Interlock Input

The Interlock input is a TTL-level, 0–5.5 V, BNC connection. When the interlock is in its normally high state, the high voltage connectors are disconnected from the power supply by relays, and the internal high voltage supply is turned off. Applying a TTL low signal (<0.2V) to the interlock enables the high voltage supply and connectors. You can also connect the supplied 50-Ω terminator ([Figure 10](#)) to the Interlock Input to enable the high voltage functions.



Figure 10. 50-Ω BNC Terminator for Interlock 1.

RS-232 Serial Port Connector

The RS-232 connector allows the HVS448 to receive programming and instructions from a host PC, via the included cable. A USB-to-serial adapter (available from LabSmith) can be used to allow communication via a USB port on the host computer.

High Voltage Connections

Each channel of the HVS448 includes a single high voltage connector. Each channel can function as a voltage/current source or monitor, so this connector acts as either an input or output, depending on the function of the channel at any time.

The maximum voltage and current of each HVS448 channel varies by model (see [Table 1](#)). Examples throughout this manual will be based on an HVS448 1500V (± 1500 V, ± 12 mA). Example voltages and currents may vary depending on your model.

The total system output (all channels considered) will not exceed 12.5mA. The system will automatically shut down the high voltage supplies if this condition occurs. Each channel's maximum and minimum voltage and current can be limited in software to protect external devices and equipment. Under no condition will the voltage or current at the channel connection exceed these software limits.



WARNING: *DO NOT connect or disconnect cables from an enabled high voltage channel. Doing so could result in personal injury and/or damage to the HVS448 or external equipment. You must disable the high voltage connectors or turn off the HVS448 before plugging or removing equipment. Use the Disable HV button*



on the toolbar or select Actions> Disable High Voltage from the dropdown menu tab.

Digital Connections

Four digital inputs and four digital outputs (BNC connectors, TTL-level, (0–0.2 V: low; 2.4–5.5V: high) allow communication with external equipment.

The inputs be connected to interlock switches, sensors or the outputs of other channels. Inputs are pulled high through a 20KΩ resistor; therefore, when open the TTL input reads high, when short-circuit it to read low. This allows passive testing of input triggers. If a TTL voltage is applied, the input can be forced high or low.

Outputs might be connected to limit switches, indicator lights, monitoring equipment, alarms, or to the inputs of other channels.



WARNING: *Do NOT connect a high voltage channel to a digital input or output. While the digital inputs and outputs are protected to 115 VAC, connecting these different voltage levels could damage the HVS448 and/or any connected equipment.*

Disable/Enable HV



Click the **Disable High Voltage** toolbar button to disable the high voltage connectors, letting them float to whatever voltage is driving them, regardless of the present sequence or settings. It is a quick way to place your system in a “safe” state.



Click the **Enable High Voltage** button to return to regular operation. This setting is nonvolatile and is restored upon power-up.

You can also disable/enable individual channels from the HVS front panel, and from within the Sequence software (See [Channel Control Panels](#)).

Recalibrating the Channels

To ensure the highest accuracy and performance, the HVS448 should occasionally be recalibrated in normal use and must be recalibrated following a firmware upgrade, which overwrites all the internal calibration data. For operations of low current where current accuracy is critical, recalibration should be performed before each experiment. For these operations, the HVS448 should be powered up and left on for a minimum of 30 minutes prior to recalibration.

1. Turn off the HVS448 and disconnect everything from the high voltage outputs.
2. Enable the high voltage connections by connecting the supplied 50-Ω terminator to the Interlock Input on the back panel.
3. Connect the HVS448 to your computer and turn on the unit.

Note: For best results with low current operations, leave the unit on at least 30 minutes before calibration.

4. Run the Sequence software.
5. In Sequence, enter Online mode by clicking the  toolbar button.
6. Click the  toolbar button to enable high voltage. The “All” indicator on the HVS448 front panel LED should glow green after a few seconds.
7. In Sequence, choose **Actions>Calibration>Recalibrate**. As the system recalibrates you will hear a relay clicking inside the HVS448, and its front panel LEDs will flash red and green.
8. When all the tasks are done, choose **Actions>Calibration>Save**. The system is now recalibrated and is ready for use.



WARNING: *While the HVS is recalibrating, the high voltage channels are supplying up to the maximum voltage to the back-panel high voltage connectors. Be sure to disconnect everything from the high voltage connectors (except for the interlock terminator) on the back panel before performing a recalibration.*

Note: If after performing a calibration you wish to return to the previously saved calibration choose **Actions>Calibration> Restore Calibration**.

CHAPTER 4. SEQUENCE SOFTWARE OVERVIEW

Sequence™ software lets you set the HVS448's high voltage functions. The sourcing and sensing functions can be set manually via control panels for each channel. The Sequence programming environment lets you create sequences that automatically control and change the voltage settings. Sequence wizards help simplify the process of defining sequences.

Field Upgradeable Firmware

The firmware in the HVS448 is field upgradeable, allowing access to the latest capabilities as they become available. To upgrade the firmware:

1. Download the latest version of Sequence from www.labsmith.com or from a link sent to you by LabSmith.
2. Turn off the HVS448 and disconnect everything from the high voltage outputs.
3. Connect the HVS448 to your computer and turn on the unit.
4. Run the Sequence software.
5. Enter Online mode by clicking the  toolbar button.
6. Choose **Actions>Update Firmware>Update All**. Click **OK** to continue.
7. The firmware will be updated—this process takes several minutes. Do not disconnect the power or the serial cable during the update.
8. When all the tasks are done, the system is automatically recalibrated to complete the system upgrade.
9. To verify the firmware version, go to **Action>Get Versions**. The firmware version number will be displayed in the status bar. The unit must be Online for the firmware version to be displayed.

Sequence Files

A Sequence file maintains the last manual settings for each channel, as well as all of the programming for a single automated sequence. Sequence files have a .seq extension.

To open a Sequence file, choose **File>Open**, then locate the file on your hard drive. You can also choose from recently opened files at the bottom of the **File** menu.

When you turn on the HVS448, all of the high voltage channels act as "Inputs;" that is, they float to the driving voltage (if no external source is supplied, then they will indicate near 0 V). When you open a Sequence file, the channels *remain* inputs. Click the **Update All** button at the bottom of Sequence's right pane to set all of the channels to the last selected settings.

Online and Offline Operation

Sequence functions in either “on line” or “off line” modes. When on line, Sequence communicates with the HVS448, allowing control and monitoring from a PC. When off line, Sequence does not communicate with the HVS448, but it does record your settings and programming which can later be stored in the HVS448.

By default, on startup, Sequence is off line. To switch between Offline and Online modes,



click the toolbar button. The mode is reported in the status bar at the bottom of Sequence’s main window. When Sequence is instructed to go on line, it automatically programs the HVS448 with the manual power supply settings.

You can also put the HVS448 online by going to **View>Option>Communications**. When the correct serial port is selected the HVS448 will automatically toggle to on line mode.

Sequence Software Environment

Figure 11 shows the Sequence main window.

This standard Windows interface includes menus and toolbars to access virtually all functions. These options will be discussed throughout the body of the text. A complete list of menu and toolbar options is also available in [Appendix 1](#).

The left pane lets you navigate through the manual controls for each high voltage channel, as well as through the various parts of an automated sequence. You can collapse or expand levels to access the settings you need while hiding extraneous functions. To expand a level, click the “+” button to the left of that item; to collapse a level, click the “-” button.

Click on an item in the left pane to view related settings in the right pane:

- **High Voltage Power Supply/Monitor** opens the **Power Supply Control Panel**, an overview of all eight channels’ manual settings
- **Channel A–H** opens the manual settings for a particular channel
- **Sequencer** opens the **Sequencer Control Panel**, an overview of the current sequence’s settings
- **Sequencer A–H** opens the sequence for an individual channel
- **Step A–H** shows a step within a sequence
- **Programmable Trigger Logic** lets you define the logic of the signal sent to the digital input.

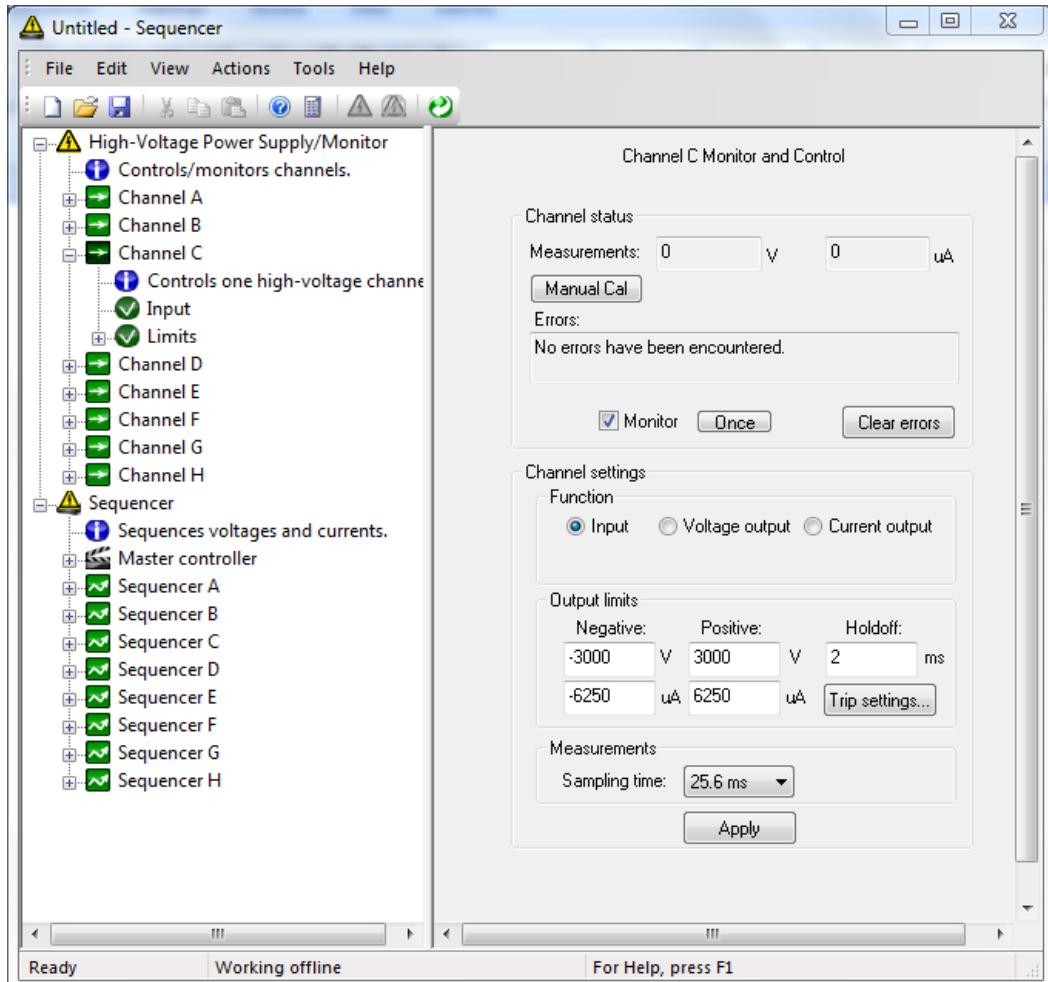


Figure 11. Sequence Main Window, showing navigation in the left pane and channel settings on the right.

Monitor Settings

Monitoring channel measurements and status will help you interpret the state of the HVS channels. Choose **View>Options>Monitoring** to control how voltage, current and sequence status will be monitored ([Figure 12](#)):

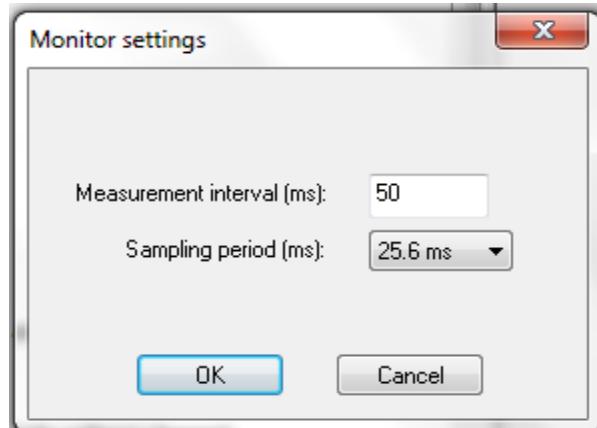


Figure 12. Monitor Settings.

- **Measurement interval** is the frequency with which measurement values are updated in milliseconds. The measurement interval must be an integer in the range of 1 to 1000000000. The new measurement interval will be applied the next time Sequence starts. To apply the new measurement interval immediately, select the new interval, click **OK**, set the software to monitor **None** (see [Manual Power Supply Control](#)) then reselect the channels to monitor.
- **Sampling period** is the default period over which measurements are integrated in the HVS. The default period is 25.6 ms; you can also select a shorter sampling time from the drop-down list.

Note: During high speed measurements the added processing required for monitoring measurements may add jitter to your timing. Use monitoring thoughtfully if you are performing high-speed or timing sensitive experiments.

CHAPTER 5. HIGH VOLTAGE POWER SUPPLY: MANUAL MODE

At its heart, the HVS448 is as a multi-channel high voltage power supply.

In “manual” mode, you set power supply settings for each channel through the Sequence software, or through other driver software such as LabView.

Channel Control Panels

Click on a channel in the left pane of the Sequence window to view its control panel (*Figure 13*).

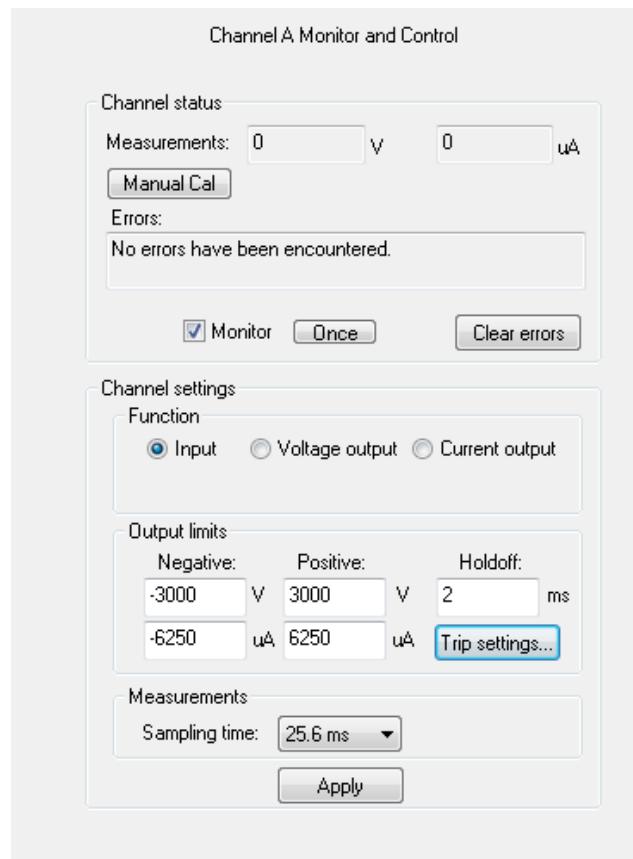


Figure 13. Channel A’s High Voltage Control Panel.

Note: If the channels are not visible in the left pane, double-click **High Voltage Power Supply/Monitor** to expand that section and reveal the eight channels, A–H. Then, click on a channel to display its settings.

Channel Status

Monitoring provides feedback on the channel's status and measures its present voltage and current values. Click **Once** to monitor and report the voltage and current. Select the **Monitor** check box monitor status and values continuously.

Note: The default monitor setting is governed by the Monitor Settings dialog. To set the default, choose **View>Options>Monitoring** (see [Monitor Settings](#)).

A channel can monitor either its internally-generated voltage or current, or the voltage/current of an external, connected signal. When the channel's setting is **Input**, the external values will be read. When the channel setting is **Voltage Output** or **Current Output**, the internal values will be measured.

Channel Settings

As mentioned earlier, any channel can:

- function as a high voltage voltmeter with 200 M- Ω input impedance
- supply voltage while monitoring current
- supply current while monitoring voltage.

Select **Input** to use a channel as a voltmeter for an incoming signal. The **Target Value** box will disappear. Click **Apply** to record the change.

Select **Voltage Output**, then enter the target voltage to set the channel to supply a regulated voltage.

Select **Current Output**, then enter a target current to regulate an output current.

After selecting the channel function click **Apply** to initiate the new function.

Channel Output Limits

To protect external equipment, you can set hard limits for the current and voltage on each channel. Enter lower and upper Output limits for voltage and/or current.

These limits are particularly useful in automated mode, to ensure that the output voltage/current stays within bounds regardless of instructions from the sequence. If an out-of-bounds condition occurs, the channel will trip (i.e., will float to the controlling voltage) or clip (i.e., will be forced within the limit values), depending on the channel's trip-handling settings (see [Error Reporting/Trip Settings](#)) below.

Note: The maximum output of the HVS448 is 12.5 mA. If the total of the current settings across all channels exceeds this amount, the overload circuit will trip, and the outputs will temporarily become inputs (i.e., they float to the driving voltage).

The **Holdoff** value is the duration that the channel must stay beyond the voltage/current limits in order for the trip settings to take effect.

Error Reporting/Trip Settings

The HVS448 channels continuously check the validity of their settings. In response to invalid conditions, the channels can trip, clip, or ignore different types of errors. Each channel can have different settings. If Monitoring is turned on for a channel, any problems on that channel will cause a message to appear in the **Errors** field.

Click the **Trip Setting** button to choose the error responses for a channel. The **Becomes an Input on** dialog box will be displayed (see **Figure 14**).

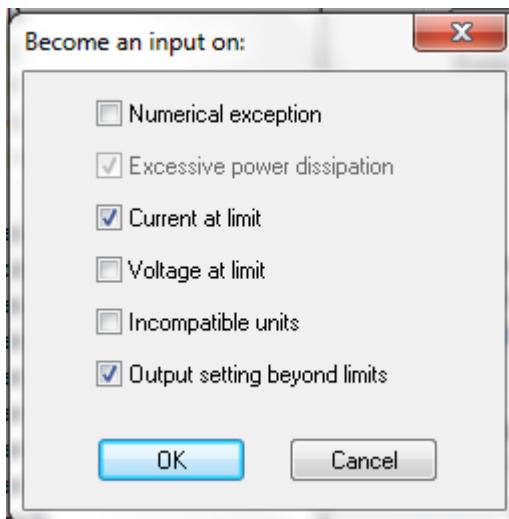


Figure 14. Channel Error Handling Dialog Box.

Table 2 describes the potential errors, the potential responses should those errors occur, and the message that will appear in the Errors field if the condition occurs.

Table 2. Error Descriptions.

Error	Meaning	If box is checked	If box is not checked	Error Message
Numerical Exception	Overflow/underflow in a mathematical calculation (e.g., square root of a negative number attempted)	Channel becomes an input (i.e., floats to driving voltage)	Error is ignored	"A numerical exception occurred."
Current at limit	The current exceeds the channel limits	Channel becomes an input	Clip to current limit	"The current exceeded the limits."
Voltage at limit	The voltage exceeds the channel limits	Channel becomes an input	Clip to voltage limit	"The voltage exceeded the limits."
Mismatched units in calculations	A mathematical operation was attempted on numbers with incompatible units (e.g., a	Channel becomes an input	Error is ignored	"Calculations were performed on numbers having inconsistent units."

Error	Meaning	If box is checked	If box is not checked	Error Message
	voltage added to a time)			
Output setting beyond limits	The output of the power supply is exceeded	Channel becomes an input	Clip to current or voltage limit	"The supply was instructed to regulate beyond its prescribed limits."

Two other fault conditions are monitored. You don't have the ability to change these behaviors:

Fault	Meaning	If box is checked	If box is not checked	Error Message
Temperature limit exceeded	The sensor in the HVS indicates excessive heat	Channel becomes an input	Channel becomes an input (cannot ignore this fault)	"The supply exceeded its safe operating temperature."
Interlock open (low)	The interlock input is set to High.	Channel becomes an input	Channel becomes an input (cannot ignore this fault)	

Manual Power Supply Control Panel

We have just seen that the settings for each channel can be set manually using the individual channel control panels. An overview of all channels can also be viewed. Click on **High Voltage Power Supply/Monitor** in the left pane to open the **Manual High-Voltage Power Supply Controls** dialog box.

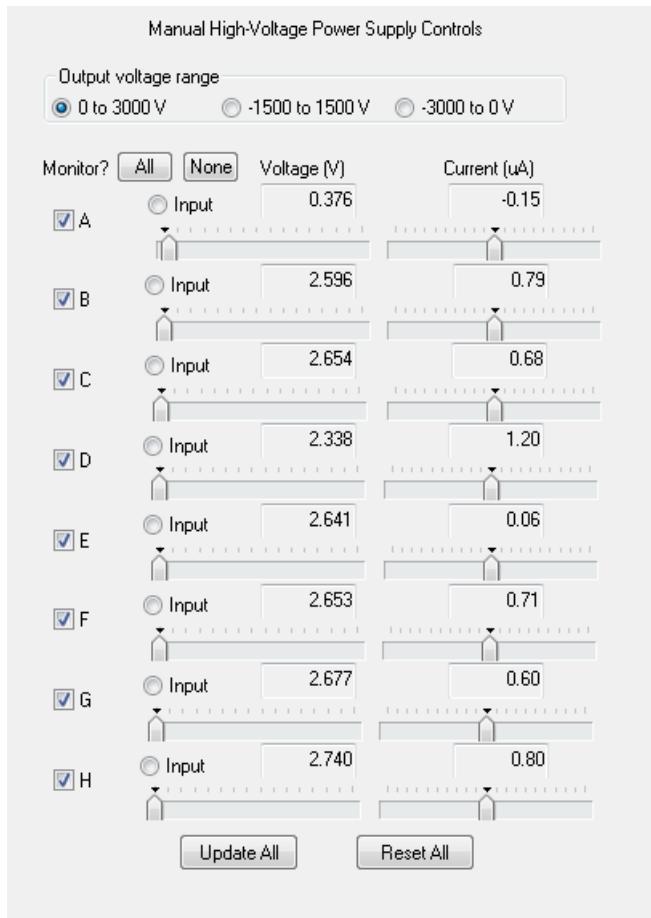


Figure 15. Manual High Voltage Power Supply Control Panel.

Overall Settings

At the top of the control panel, set the **Output Voltage Range** to a value that makes sense for your current experiment. The voltage and current sliders for each channel will adjust to reflect your choice.

Update All synchronizes the data from all channels, reads the calibration data, updates all channel settings, and updates the monitored values once for all channels. This function also appears on the Sequencer Control Panel.

Reset All makes all channels inputs, clears all error flags, and stops a sequence if one is running.

Monitoring

Select the **Monitor?** check box next to each channel to display that channel's real time Voltage and Current values. You can also click **All** to monitor all channels, or **None** to turn off all monitoring.

Channel Settings

You have the following controls for each channel:

- Click the **Input** button to use the channel as a voltmeter for an incoming signal. If monitoring is selected, the voltage and current boxes will show the values of the external signal.
- Click and drag the **left slider** to set a target voltage for the channel to regulate. As you move the slider, the target voltage will be displayed. You can also enter the target value in the Voltage box. You will not be able to move the slider beyond the **Voltage Limits** for the channel (as set in the *Channel Control Panels*).

If Monitoring is on for that channel, the Current slider will adjust to reflect the present measurement of current (*Figure 16*).

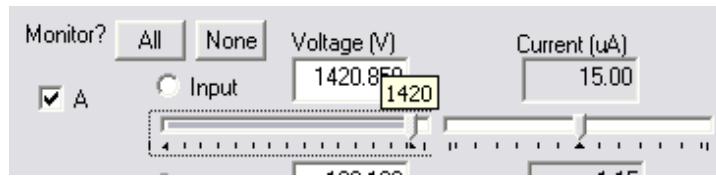


Figure 16. Click and Drag the Slider to Change Channel A's Voltage.

- Click and drag the **right slider** to set a target current for the channel to regulate. As you move the slider, the target current will be displayed. You can also enter the target value in the Current box. You will not be able to move the slider beyond the Current Limits for the channel (as set in the *Channel Control Panels*).

If Monitoring is on for that channel, the voltage slider will adjust to reflect the present voltage measurement.

Error Flags

If by changing the target Voltage value you exceed the current limits (or vice versa), the channel will trip or clip, depending on the error handling settings (See *Error Reporting/Trip Settings*).

When this happens an **Error** button  will appear to the left of the channel. Click the button to view the encountered error. A dialog box will appear (*Figure 17*). Click **Yes** to reset the channel.



Figure 17. Errors Encountered Dialog Box.

View Channel Settings in the Left Pane

For a quick overview of a channel's settings, expand the channel in the left pane ([Figure 18](#)):

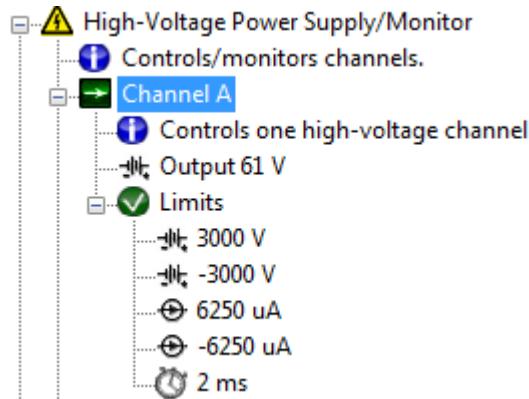


Figure 18. Channel A's High Voltage Settings Shown in the Left Pane.

The function and limits from the [Channel Control Panels](#) will be displayed. In the example above, Channel A has been set to regulate voltage at 61 V, with lower and upper limits of 3000 V and -3000 V. The current limits have been specified at 6250 and -6250 μ A. The Holdoff value is set to 2 ms.

Each channel includes a text label in the left pane, which, by default, reads "Controls one high voltage channel." You can change this text to identify the channel more specifically; for example, you may specify the device to which the channel is connected. To change the label, click the text to select it, then click again to highlight it for editing.

Using Trace Measurements to Record Data

The Trace Measurements option lets you record the voltage and current for each channel over a period of time, using the HVS as a data recorder. The data is stored to a tab-delimited file, with a .trc extension by default, which can be imported directly into a spreadsheet program for further analysis. Choose **File>Trace Measurements** to start recording; choose **File>Stop Tracing** to end the recording.

Note: You should not select or deselect channel monitoring while Trace Measurements is enabled. Doing so can lead to unexpected output.

CHAPTER 6. HIGH VOLTAGE SEQUENCING: AUTOMATED MODE

As you have seen, the HVS can function as an eight-channel, high voltage power supply and sensor. You can set its channels manually and monitor the results of those changes.

The next step is to automate the channel settings with the HVS's sequencing environment, which gives you the ability to execute coordinated, non-trivial programs at high speed. Automation is accomplished through “sequences.” A Sequence is a set of automated commands grouped into **Programs**. Programs consist of individual **Instructions**.

In automated mode you will typically perform the following steps:

- In Sequence, program the sequence steps and settings for each channel.
- Store those properties and logic to the HVS448's memory.
- Test your sequence by manually issuing control commands and examining the channel states, using the software, the HVS448's channel LEDs and/or a multimeter.
- Plug your experimental equipment to the HVS448.
- Run the experiment, using the HVS448 by itself or with the computer.

HVS448 System Architecture

The HVS448 architecture is shown in [*Figure 19*](#).

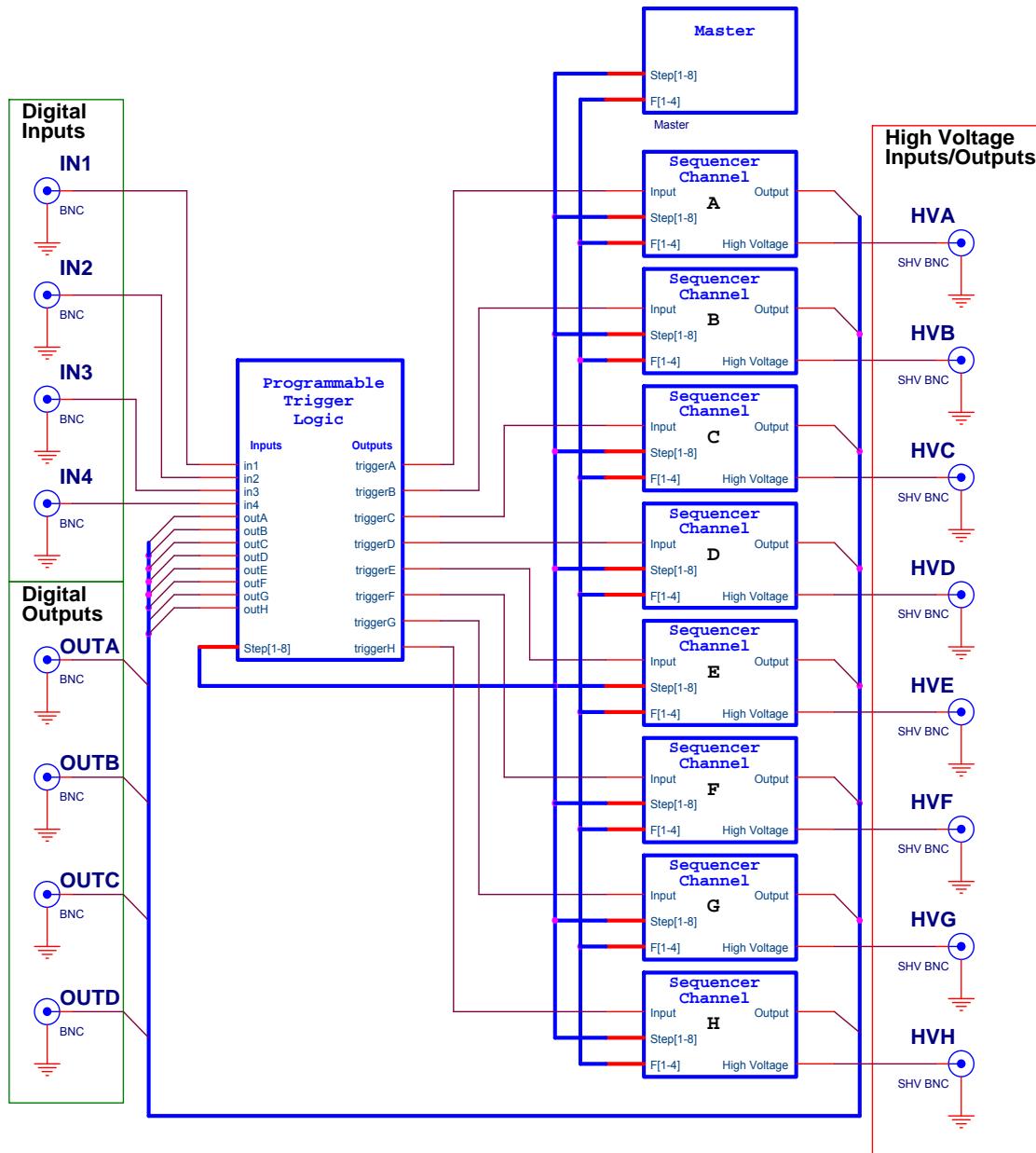


Figure 19. HVS448 System Architecture.

Master and Channel Sequencers

The highest authority in this architecture is the **Master Sequencer**. It coordinates the functions of the eight **Channel Sequencers** (A –H), which control and monitor the high voltage inputs/outputs. Unlike the channels, the Master does not control a high voltage output, nor does it trigger inputs and outputs.

The Master and Channels execute independent **Sequences** in parallel.

While the Master and Channels run independent sequences, they can also communicate via three methods:

- **Step instructions** let all channels jump simultaneously to defined points in their sequences
- **Flags**, which are variables that can be set, cleared and/or read by any channels
- **Trigger signals**, which let a particular channel respond to a combination of internal or external events.

Manual Changes During an Automated Sequence

When you run a sequence, the Sequence software will automatically change channel functions and settings. At the same time, you can continue to make manual changes through the **Channel Control Panels** and **Manual Power Supply Control Panel**. You may need to do this, for example, if you want an event to occur when Channel A reaches some yet-unknown voltage level. The HVS gives you the ability to manually adjust that voltage at the appropriate time so that you can determine that unknown level.

Once you start running a sequence, the channel states will change automatically.

Inter-Channel Communication

In all but the simplest sequences, the channels must be able to influence each other, for example, to signal an over-current condition or to move to a new step in the sequence. There are three mechanisms of inter-channel communications: step instructions, flags, and triggers.

Use **Step** instructions to switch between conceptual portions of a sequence, when multiple channels must react simultaneously.

Use **Flags** for general purpose signaling among channels. Flags can be assigned different meanings by the user at different times within a step.

Use **Triggers** to communicate with external equipment, to create “protected” communication pathways between channels, and to facilitate complicated signaling logic. Trigger logic can be different for different steps, but is constant during a step.

Step Instructions

Step instructions are interrupts which force the master and channel sequences to jump from their current programs to a prescribed one-instruction step program. Step instructions should be used where there is a clearly defined change that affects multiple channels—for example, switching from an “initialization” step to a “run” step.

A one-instruction step program will include a “Continue”, “Run” or “Exit” instruction.

A “Continue” instruction resumes the program that was running before the Step instruction.

A “Run” instruction jumps immediately to a global program label. In the example:

```
Run Initialize
```

On switching to this step, the program “Initialize” will be run.

It is good programming practice to use the “Run” instruction at least once for any step that is to be run by a user command for clarity, even when it is not necessary.

Any channel, or the master, can force a step change at any time by executing a “SwitchTo Step” instruction. All channels must switch to the prescribed Step at the end of their current instruction, regardless of current activity.

There is no way for a channel to ignore a step change. However, by using the “Continue” Step instruction the channel will resume execution following the step change.

Note: You can also manually execute a Step instruction during an automated sequence, by choosing **Actions>Run>Step A–H**, or by selecting it on the Sequence Control Panel.

The **Exit** instruction causes a channel to stop running the sequence and sets the channel to function as an input.

Orderly Step Flow

Though any sequence can perform a Step instruction, one sequence channel or the master must retain authority over step instructions at any given time to avoid multiple sequences sending simultaneous (and inconsistent) Step instructions.

A simple solution is to program only the Master sequence to issue Step instructions. While this approach does help ensure orderly movement between steps, it does not work universally, as when different channels must switch steps in response to measurements. Whatever the requirements, it is best to keep the step switching simple.

Step H: The Exit Step

Step H is reserved for ensuring a safe shutdown of the sequence. When a channel executes an “Exit” instruction, it switches the sequencer to Step H *and* does not allow any other step instructions until the master sequencer terminates the program by issuing its own “Exit” instruction (see **Exit**). If any special shutdown procedures are required by a

channel, they should be run in Step H. The default step instruction of the Master for StepH is “Exit,” causing the sequence to exit immediately at StepH. Thus, by default only, a “SwitchTo StepH” instruction has the effect of terminating a sequence.

Flags

Flags are useful for coordinating program flow within a step. The master or channels can “set” and “release” the four flags, F1 through F4, and the flags can have different meanings at different points in a sequence. Unlike step instructions, flags do not automatically produce a particular behavior; instead, individual channels can react to, or ignore, the flags.

For example, if two channels are to produce out-of-phase square waves during a step, the channels could coordinate their voltage-switching times by the use of a flag. Other channels would go about their business ignoring, watching, or manipulating this flag as desired.

The master or channel sequences set and release flags by the instructions “SetFlag X” and “ClearFlag X,”

where X = F1, F2, F3, or F4.

The state of the flag can be read by the conditional statements, “FlagSet? X” or “FlagClear? X,”

again where X is F1, F2, F3, F4. If the test is true, the next instruction will be executed; if the test is false, the next instruction will be skipped.

If two or more channels set a flag, and one releases the flag, the flag remains set. A flag is only cleared when all channels that have set it have also released it.

Triggers

Triggers are used for communication with external equipment, or when a compound condition is required to “trigger” an event in a channel sequence. A Trigger is specific to a channel, whereas any channel can set or release a flag.

To use a trigger, you first describe a logical condition that will set the trigger. This happens at the Sequence level, following a Step Instruction. Then, within the body of a program, you test whether the trigger is set.

Trigger Logic in a Sequence

Four BNC inputs, in1 through in4, accept external trigger signals. These inputs are normally high, which represents a logical FALSE trigger value. Tying an input’s voltage low sets the trigger value to a logical 1, or TRUE.

Each channel has a trigger output. The outputs of channels A, B, C, and D are connected to the back-panel BNC connectors labeled outA, outB, outC, and outD, respectively, allowing these channels to trigger external apparatus.

Each channel also has a trigger input, which you can program to be any logical comparison of signals from in1 through in4 and/or outA through outH. The default trigger logic for a channel input is:

```
trigger = FALSE;
```

meaning, “the channel’s trigger will never be set.”

Trigger logic can be more complex, as in:

```
trigger = outB and not in1;
```

which means, “this channel’s trigger will be set if Channel B sets its trigger output WHILE input in1 is FALSE.”

Trigger logic can be different for each sequence step:

```
StepA: Run Initialize;  
trigger = FALSE; //the trigger is never set during StepA  
StepB: Continue;  
trigger = in1; //the channel can be triggered off the  
external input in1, during StepB
```

Note: As in the example above, comments can be added to a sequence by placing “//” before the comment text.

If you define trigger logic for a channel in Step A it does not automatically carry over to Steps B and C; those steps will retain the default trigger logic until you change it.

Triggers isolate communication channels in a way that is not possible with flags, reducing the potential for software bugs. For example, the earlier example of coordinating channels to make complementary square waves may be better implemented using triggers rather than flags. Then another channel *cannot* accidentally change the coordination of the channels by setting the flag.

Fixed Trigger Logic for the Master Sequence

The trigger condition for the master sequence is set to

```
trigger= outGND
```

which is always a logical TRUE. Unlike the channels, the logic for the master cannot be changed. The Master can only be triggered via a **Force Trigger**. To do so, in the

Sequence Control Panel (**Figure 20**) click the Master **Force Trigger**  button. Alternately, on the HVS front panel press **Trigger**, then press **ALL**.

Trigger Set, Clear and Read in a Program

The trigger output of a channel can be set or cleared using the “Output Set” and “Output Clear” instructions.

The trigger input state can be tested using the commands “InputSet?” and “InputClear?”

If the condition is true, the next instruction will be executed; if it is false, the next instruction will be skipped.

Sequence and Program Flow

A one-instruction program executes in response to an internally generated step switch instruction (see [SwitchTo](#)), a “Run” Step command, or simply a “Run” command. This program can be only one of two instructions: “Continue” or “Run X”, where X is the name of a channel program. At the same time, the input trigger logic switches to the prescribed logic setting.

An example channel sequence could be:

```
StepA: Run Initialize;
trigger = FALSE; // the trigger for this channel is always
FALSE during StepA
StepB: Run Inject;
trigger = in1;           // the trigger responds to input in2
during StepB
StepC: Continue;
StepD Continue;
...
```

with a channel program for regulating voltage as follows:

```
Label Inject      ;
500V ;
Reg Voltage      ;
Stop ;
```

In this example, the sequence controls program flow between two logical steps: an initialization phase, and a phase in which the channel regulates voltage. At each step, the sequence calls a single program. When the sequence reaches StepB, the “Run Inject” instruction is executed. The next instruction, “Label Inject,” names the “Inject” program. That program begins to run, regulating voltage at 500 V.

Monitoring and Interacting with an Automated Sequence

Sequence’s automated mode includes an overall Sequencer Control Panel and individual Channel Sequence control panels for monitoring and interacting with the automated sequence. These are analogous to the [Manual Power Supply Control Panel](#) and [Channel Control Panels](#) in manual mode.

Sequencer Control Panel

This control panel gives you an overview of the present automated sequence. To open the Sequencer Control Panel, click **Sequencer** in the left pane ([Figure 20](#)).

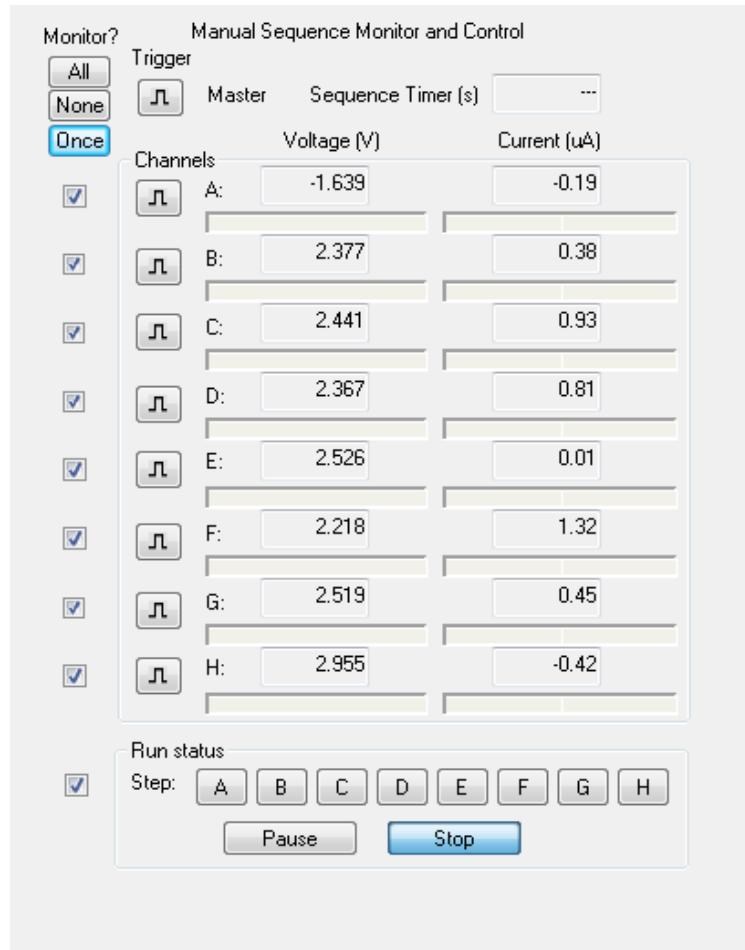


Figure 20. Sequencer Control Panel.

Many of these controls will be familiar from the Manual Power Supply Control Panel. You can check the **Monitor** box next to any channel to view the present voltage and current settings. Additionally, the bars beneath the Voltage and Current boxes show the present values as a percentage of the overall voltage or current range. Click **All** to monitor all channels; click **None** to turn off all monitoring.

 Click the **Trigger** button to force the trigger input for a channel. This is a useful way to interact with the automated sequence. For example, if you must manually adjust some portion of the experiment during a run, you could include an **Await** command that will resume the automated run once you've forced the channel trigger.

You can also use an **Await** command to Force Trigger the master sequence (since it has no trigger logic, this is the only way to trigger the master).

You can manually issue a **Step** instruction by clicking one of the **Step** buttons at the bottom of the panel. The button will appear depressed to indicate the present step.

Click **Pause** or **Stop** to interrupt the current program.

Check the bottom-most **Monitor** box to automatically show the present step and status of the sequence. The Step buttons will depress to indicate the current step of the sequence.

Update All synchronizes the data from all channels, reads the calibration data, updates all channel settings, and updates the measurement monitored values once for all channels. This function also appears on the [Manual Power Supply Control Panel](#).

Channel Sequence Control Panels

The Channel Sequence panels ([Figure 21](#) shows one for Channel A) let you monitor the progress of each particular channel during an automated run. Click Sequencer A–H in the left pane to open the control panel for that channel.

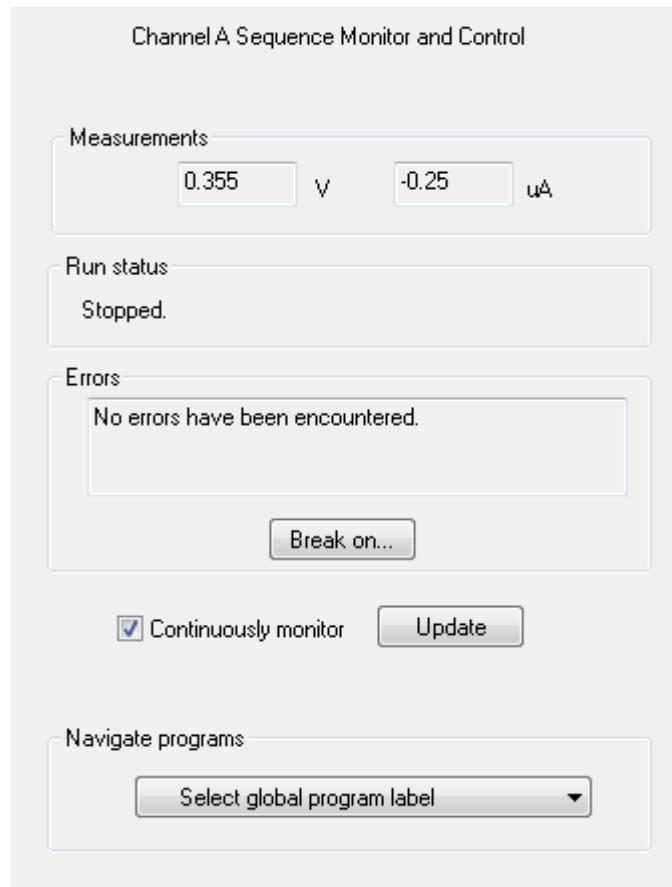


Figure 21. Sequencer Channel A Sequence Monitor and Control Panel.

If Monitoring is on for this channel (as set in the [Sequencer Control Panel](#)), the **Measurements**, **Run Status**, and **Errors** fields will show the channel's last measured voltage and current values, status, and encountered errors. Click **Update** to refresh the values. Check the **Continuously Monitor Measurements** box to update the values on a continual basis.

The **Navigate Programs** drop-down menu shows the programs for the channel. Select a program to view its contents in the right pane.

Error Reporting/Break Settings

The HVS448 channels continuously check the validity of their settings. In response to invalid conditions, each channel can stop its sequence. You can control which errors will stop a sequence by clicking the **Break on...** button. The **Stop the sequence on:** dialog box will be displayed (*Figure 22*). These controls are similar to the **Error Reporting/Trip Settings** controls in manual mode, but here the settings determine whether a channel sequence will be stopped (rather than determining when a channel will clip or trip). If selected, an error will cause an “Exit” instruction to be executed by the sequencer. See [Exit](#).

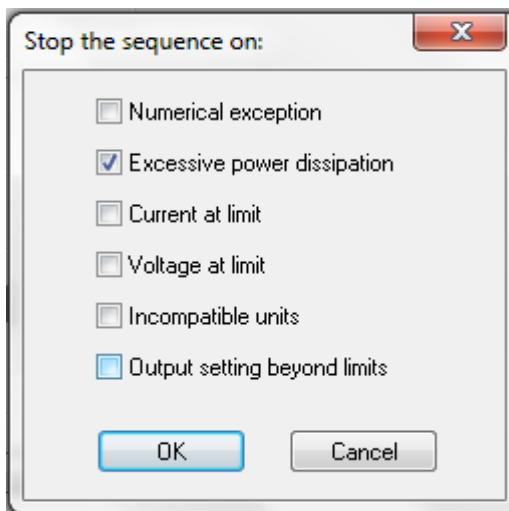


Figure 22. "Stop the sequence on:" Dialog Box.

The possible errors conditions are:

- **Numerical Exception:** Overflow/underflow in a mathematical calculation (e.g., the square root of a negative number was attempted)
- **Current at limit:** The current exceeds the channel limits
- **Voltage at limit:** The voltage exceeds the channel limits
- **Incompatible units:** A mathematical operation was attempted on numbers with mismatched units (e.g., a voltage added to a time)
- **Output setting beyond limits:** The output of the power supply is exceeded

All sequences will be suspended automatically if the internal temperature of the HVS exceeds its limits, or if the Master Interlock is released (see [Master Interlock Input](#)).

The Programming Interface

The Sequence interface is designed to follow the same hierarchy as the software: sequences consist of steps and programs; programs consist of instructions.

In the left pane, double-click **Sequencer** to expand and view the master and channels controls ([Figure 23](#)).

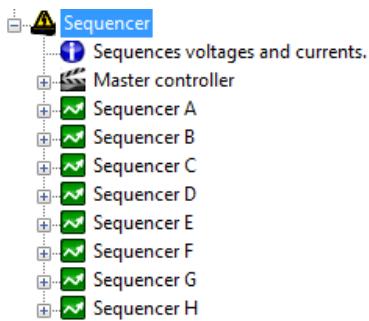


Figure 23. Sequencers Shown in the Left Pane.

View the steps of a particular sequence by expanding it; expand a step to see its step instructions and trigger logic ([Figure 24](#)):

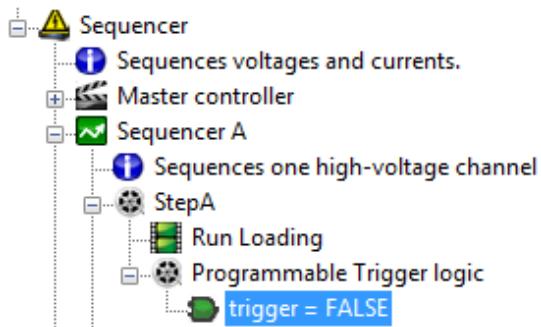


Figure 24. Steps Shown in the Left Pane.

Expand the **Programs** list to view the programs available for a sequence; expand a particular program to view its instructions ([Figure 25](#)):

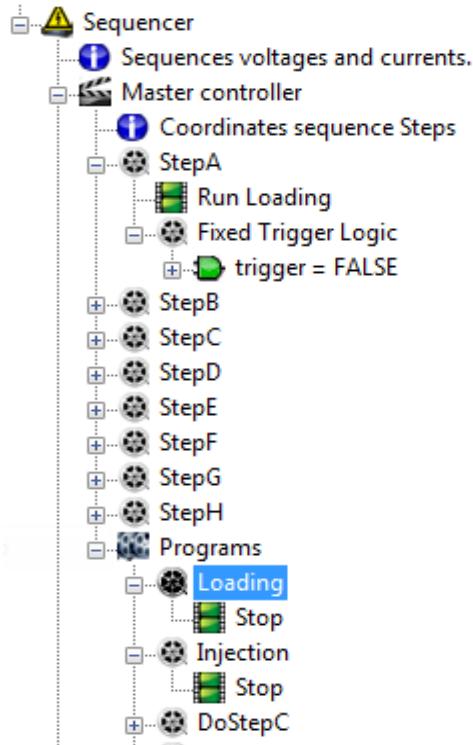


Figure 25. Programs and Instructions Shown in the Left Pane.

Editing Sequences

As we've seen earlier, the Sequence Wizards give you a simple interface for writing sequences (see [Using the Sequence Wizards to Create a Sequence File](#)). Once a sequence is created, you have three options for editing a sequence or program:

- Edit an item directly it in the left pane
- Click on an item in the left pane, then edit its text in the right pane
- Use the **Programming Keypad** to enter commands.

View and Edit Instructions in the Left Pane

When you edit a sequence or program in the left pane, Sequence will add the correct syntax, saving you programming and debugging time.

To edit an instruction in the left pane, click to select it, then click again to edit it ([Figure 26](#)).

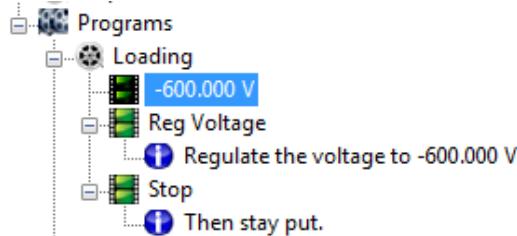


Figure 26. Editing a Program Instruction in the Left Pane.

To add a new instruction, right-click on an instruction within that program. From the drop-down menu that appears, choose **New>Instruction**. A new instruction will appear at the end of the program (Figure 27).



Figure 27. Adding a Program Instruction in the Left Pane.

Once the new instruction is added, click it twice to edit it.

To delete an instruction, right-click on it, then choose **Delete**.

Several objects in the left pane include text messages to help identify them. You can alter these messages by double-clicking on the text in the left pane (Figure 28).

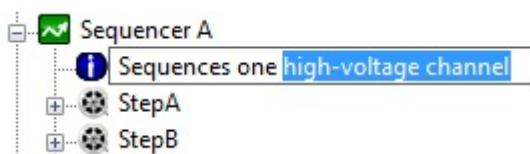


Figure 28. Editing a Label in the Left Pane.

View and Edit Instructions in the Right Pane

Experienced programmers often choose to edit straight text in the right pane. Care must be taken to use correct syntax, however.

When you click on an item in the left pane, the details for that item will appear on the right:

- Click on a **Step** to view and edit its one instruction (Run, Continue or Exit)
- Click **Programmable Trigger Logic** to view and edit the logic for that channel and step
- Click a **Program** to view and edit its contents.

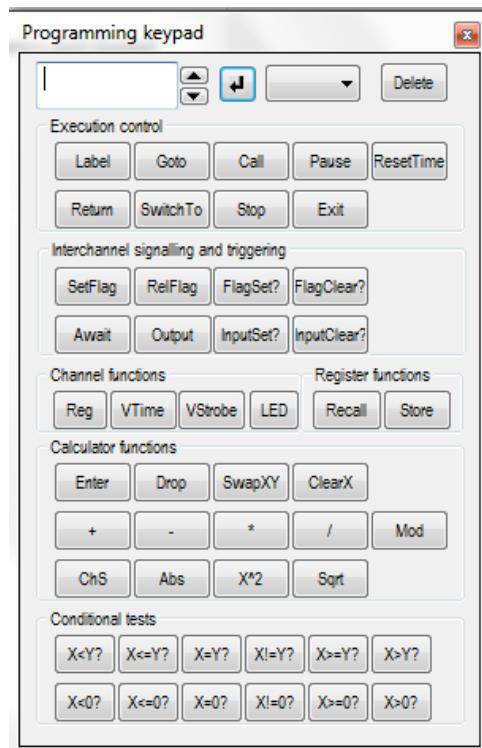
Once you've displayed an object's contents in the right pane, you can use standard text editing (including cut, copy and paste) to alter the code ([Figure 29](#)).

```
Separate ;  
          0 V ; Running buffer source channel  
Reg      Voltage ;  
Stop ;
```

[Figure 29. Editing a Program in the Right Pane.](#)

Edit with the Programming Keypad

The Programming Keypad ([Figure 30](#)) is a very fast way to access all the different sequence and channel program commands. To open the keypad, click the  toolbar button.



[Figure 30. Programming Keypad.](#)

To use the keypad, open the text of a sequence or program in the right pane. Place the cursor where you want to add a command, then click the required instruction on the keypad. The instruction will appear in the box in the upper left of the keypad. Click the  button on the keypad to repeat the last instruction. The instructions are described in detail in [Chapter 7. Channel Program Instructions](#).

Some instructions require an argument; for example, **Goto** requires a label for the destination. When you select such a function, a list of available arguments will appear in a pop-up in the upper right corner of the keypad ([Figure 31](#)). Choose one, and the entire instruction will be added to the program.



Figure 31. Entering an Instruction and Argument Using the Keypad.

To delete an instruction, select it in the left pane, then click the **Delete** button on the keypad. You can also highlight the text in the right pane and cut/delete it.

Error Checking and Parsing

Each time you enter an instruction, Sequence checks the syntax for you. If an error is found, the icon for the errant instruction will appear in red in the left pane. The program containing that instruction will also appear in red, as will the sequence containing the program, and the Sequence icon itself, to help you quickly spot the problem.

In the example in [Figure 32](#), the same label appears twice in the same program. Sequence has flagged this as an error.

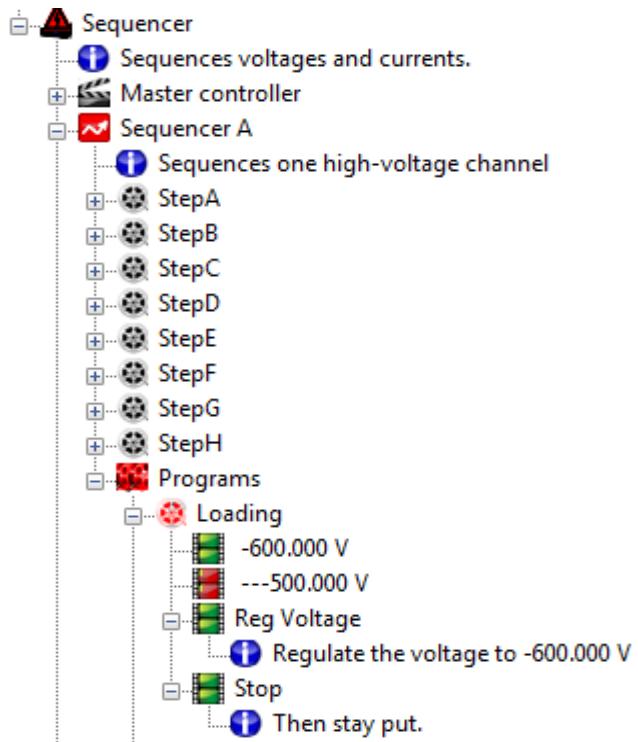


Figure 32. An error like this repeated label is flagged in red all the way up the hierarchy.

Though Sequence can help you debug syntactical errors, it cannot spot logical flaws in the way your sequence is structured. Good programming practices should be applied to limit the number of errors and the length of the debugging process.

CHAPTER 7. CHANNEL PROGRAM INSTRUCTIONS

HVS448 programming is relatively easy to learn: there are only about 40 instructions to understand. The following sections detail the instructions that are used in the Master and Channel Sequencers, which are referred to as “channels” for brevity. The functions are listed as they appear on the Programming Keypad (see [Edit with the Programming Keypad](#)). The programming language can be bypassed by using the [Simple Sequence Wizard](#) and [Complex Sequence Wizard](#) as described in [Chapter 2. Getting Started](#). An example of a sequence written in the Simple Sequence Wizard is given in [Appendix 1. Menus and Toolbars](#).

Nomenclature

Square brackets ([]) indicate a required argument. Curly brackets({}) contain comma (,) separated lists and/or dash (–) separated ranges of allowable arguments. The brackets, commas, and dashes should not appear in the instruction, for example,

```
Store [{A–M, Voltage, Current, Time}]
```

means that Store A, Store B,...Store M, Store Voltage, Store Current and Store Time are the allowable Store statements.

Execution Controls

Execution control instructions suspend program execution or jump execution elsewhere in the program.

Label

Usage:

```
Label [unique case-sensitive program label]
```

The label instruction is used to name a program or code segment. These names are used by the program commands Goto and Call, and the Step instruction Run. The instruction has no further effect.

Syntax:

Program labels can start with and contain any character except white-space characters and semicolons (;). There is no restriction on length.

Notes:

Global Labels

All program labels that start with a capital letter are globally known by all programs of the particular channel. No two global labels can be the same for any of the programs of a channel.

Local Labels

Program labels that start with a lower-case letter are local to the particular program. Local labels allow a programmer to use simple labels like 'a' and 'loop' without worrying that the same names are used in routines in other program listings. A local label cannot be repeated within the same for a program listing.

Example:

```
Label Inject      ; Defines a global program label "Inject"
```

Goto

Usage:

```
Goto [unique case-sensitive program label]
```

The Goto instruction causes the program to "jump" to the labeled program or code segment. It has no other effect.

Syntax:

The label must have a valid syntax and be recognized. See the documentation of the Label instruction for more information.

Example:

```
Goto Inject      ; Jumps to the global program label "Inject"
```

Call

Usage:

```
Call [unique case-sensitive program label]
```

The Call instruction causes the program to "jump" to the labeled program or subprogram. Unlike Goto, the program or subprogram can resume executing from the instruction immediately after the Call instruction, by means of the Return instruction. Call is generally used instead of Goto to execute subroutines or subprograms.

Syntax:

The label must have a valid syntax and be recognized. See the documentation of the Label instruction for more information.

Notes:

When the Call instruction is executed, the current program position is added to a 16-level stack. When the Return instruction is executed, this position is popped from the stack and the program resumes from the instruction after that point. A subroutine, therefore, can call another subroutine, which can call another, etc., up to 16-calls deep. If Call is executed when the stack is completely full, the bottom-most return location will be overwritten, so that when the seventeenth Return is executed, it will return to an indeterminate (invalid) location.

Normal programs are unlikely to approach the stack capacity. However, a user may choose to write programs that do not always return from calls a variety of reasons. In such cases, the "unused" call-stack entries will pile up on the stack, eventually being

overwritten by new calls. Overwriting these unused entries is not a problem, provided the user never intends to return from those calls.

Example:

```
Call Inject ; Calls the program or subprogram Inject
```

Pause

Usage:

```
Pause ;
```

The pause instruction takes no arguments causes the program to be suspended in its current state for a duration equal to the product of the time-base period (0.100 ms) and the value in the X register. It has no other effect.

Notes:

The Pause command actually measures the duration since the start of the program, a ResetTime instruction, an Await instruction, or a previous program pause. This allows a user to ignore the execution time of any instructions between these timing commands provided it is less than the specified duration.

Example:

```
... Do something that takes < 12.5 ms
12.5 ms ;
Pause ; pause until the pause counter reaches 12.5 ms
... ; execute instructions that take less than 1.25 ms
1.25 ms ;
Pause ;
... ; this instruction occurs precisely 1.25 ms after the
previous
... ; Pause completes
```

ResetTime

Usage:

```
ResetTime
```

The ResetTime instruction is used to clear the Time register. It can be used, for example, to synchronize clocks following a SwitchTo StepX instruction by having all channels execute a ResetTime instruction at the start of the new step.

Example:

```
...
event that takes > 12.5 ms
12.5 ms
Pause ; pause immediately completes because the pause timer
... ; exceeds 12.5 ms
```

```
...      ; (Do something that takes > 12.5 ms)
ResetTime
12.5 ms
Pause ; pause completes 12.5 ms after the ResetTime command
```

Notes:

The value written to the Time register is the elapsed time in 1/10000ths of a second since the last ResetTime instruction. When a sequence begins, or when a Step instruction is executed, the ResetTime instruction is automatically called, allowing channels to be synchronized. The Time register is also reset at the end of an Await command. If you Stop or pause a sequence, the Time register continues to advance.

Return

Usage:

```
Return
```

This function returns to the instruction following the last Call instruction.

Notes:

Executing a Return for which there is no valid Call produces an indeterminate (invalid) result. See the discussion for the Call command.

SwitchTo

Usage:

```
SwitchTo Step {A-H}
```

This command causes all channels immediately to execute the Step instruction for the prescribed step.

Notes:

All channels can execute this instruction, although it is good programming procedure to have only one channel, usually the Master, execute this instruction and thus control program flow. Having multiple channels execute this instruction can be confusing and can produce unpredictable results if channels simultaneously execute the instruction. The exception to this rule is the instruction SwitchTo StepH, which *always* properly switches execution to Step H regardless of other simultaneous instructions. Step H can therefore be reserved to handle an exceptional case or when a program is to terminate, such as when a channel detects a problem that requires an immediate shutdown or a channel executes an Exit instruction. See the description of Exit for more information.

Stop

Usage:

```
Stop
```

The Stop instruction takes no argument and causes the channel to pause indefinitely in its current state until a user command or Step Run instruction.

Notes:

The Stop instruction terminates one of two ways: by a user command or by a step change initiated by another channel (see the SwitchTo instruction documentation). If the Step instruction is Continue, the program will remain stopped.

Example:

```
...  
Stop ; The program waits here for a Step change or user  
command.
```

[Exit](#)

Usage:

```
Exit
```

The Exit instruction takes no argument and behaves differently for channels and the master. The channels cannot directly force the sequence to terminate. When a channel issues an Exit instruction, the channel executes a SwitchTo StepH instruction and forces the sequence to remain in Step H until the sequence terminates. Recall that Step H can be reserved for any procedures that are needed to terminate a sequence safely. Only the master has the ability to terminate a sequence and the Exit instruction is how this is accomplished. By default, the step instruction for the master's StepH is Exit. Thus when a channel executes Exit, the master immediately terminates the program by itself executing Exit.

Notes:

It is not necessary to terminate a sequence using an Exit instruction. A sequence can remain in an infinite loop or stopped until a user intervenes. When a sequence terminates, all the channels automatically become inputs. To avoid this behavior at the end of a sequence, simply stop or loop each channel in the desired state. Exit instructions may be best reserved to handle exceptional cases and abnormal sequence termination.

Example:

Channel A–H: ...

```
Exit ; The program switches to and forces Step H
```

Master:

```
Exit ; The program for all sequencers terminate immediately.
```

Inter-channel Signaling and Triggering

Flags and triggers can be set and released to control program execution, in response to signals from other channels or external equipment.

SetFlag

Usage:

```
SetFlag {F1, F2, F3, F4}
```

This instruction sets the specified flag. The flag can be released by executing the RelFlag instruction.

Notes:

More than one channel can set a flag.

RelFlag

Usage:

```
RelFlag {F1, F2, F3, F4}
```

This instruction releases the specified flag. a flag is only “cleared” when all channels that set a flag have, in turn, released the flag. This instruction has no other effect.

FlagSet?

Usage:

```
FlagSet? {F1, F2, F3, F4}
```

This instruction tests whether the specified flag is set. If it is set, the next instruction is executed. If it is not set, the next instruction is skipped. This instruction has no other effect.

Example:

```
FlagSet? F1 ; Test whether F1 is set
Goto HandleF1Set ; If it is do this
... ; If it is not, do this
```

FlagClear?

Usage:

```
FlagClear? {F1, F2, F3, F4}
```

This instruction tests whether the specified flag is clear. If it is clear, the next instruction is executed. If it is not clear, the next instruction is skipped. This instruction has no other effect.

Example:

```
FlagClear? F1 ; Test whether F1 is clear.
Goto HandleF1Clear; If it is do this.
... ; If it is not, do this.
```

Await

Usage:

```
Await [{F1, F2, F3, F4, NewStep, Trigger}]
```

This instruction pauses until the specified event:

- F1-F4: Pause until the respective flag is set. If the flag is already set, complete the instruction immediately.
- NewStep: Pause until a step instruction occurs. The instruction will complete only after the execution of a “Continue” instruction.
- Trigger: Pause until the channel input transitions from low to high.

When the Await command completes, it resets the Pause timer.

Example:

```
...
Await Trigger      ; Resume execution after low-to-high
transition of input
10 ms            ;
Pause            ;
...              ; Execute these instructions 10 ms after the
trigger.
```

Output

Usage:

```
Output {Set, Clear}
```

This instruction sets the trigger output state. This instruction has no other effect.

Example:

```
Output Set ; Set the trigger output high for 10 ms
10 ms
Pause
Output Clear      ; Now set the trigger output low.
```

InputSet?

Usage:

```
InputSet?
```

This instruction tests whether the trigger input is set. If it is set, the next instruction is executed; if it is not set, the next instruction is skipped. This instruction has no other effect.

Example:

```
Label a
InputSet? ; Loop while the trigger input is set
```

```
Goto a
...
; Get here when the trigger input is clear
```

InputClear?

Usage:

```
InputClear?
```

This instruction tests whether the trigger input is clear. If it is clear, the next instruction is executed; if it is not clear, the next instruction is skipped. This instruction has no other effect.

Example:

```
Label a
InputClear?      ; Loop while the trigger input is clear
Goto a
...
; Get here when the trigger input is set
```

Channel Functions

Channel functions determine how a channel will perform and how its status will appear on the front panel of the HVS448.

Reg

Usage:

```
Reg {Voltage, Current, None}
```

The instruction Reg sets the regulation mode of the channel:

- Reg Voltage: Regulate the voltage to the value stored in the X register
- Reg Current: Regulate the current to the value stored in the X register
- Reg None: Do not regulate. The channel is a high-impedance input.

Example:

```
100V
Reg Voltage      ; Regulate the output voltage to 100 V
1s
Pause          ; Wait 1 s
200V
Reg Voltage      ; Now regulate the output to 200 V
1s
Pause          ; Wait 1 s
-1mA
Reg Current      ; Now regulate to -1 mA (1 mA into the
channel)
```

```

1s
Pause
Reg None ; Make the channel a high-impedance input

```

LED

Usage:

```
LED {Green, Red, Off}
```

The instruction LED sets the state of the front panel LED indicator for the channel:

- LED Green: Display a green indicator light.
- LED Red: Display a red indicator light.
- LED Off: Turn off the indicator light.
- LED Auto: (Default) The LED setting is automatic (the same as in manual mode).

For channels, in automatic mode, the LED is red when current is flowing out of the channel and green when current is flowing into the channel. For the master, the LED is green when the high voltage outputs are enabled and red when the outputs are disabled, for example by user command or a trip.

The indicator LED is useful for showing the status of a channel.

Example:

```

1mA
Recall CurrentMeas
X>Y? ; If the current is > 1 mA
LED Red ; glow red.
X<=Y? ; Otherwise,
LED Green ; glow green.

```

Register Functions

Register Functions let you temporarily store and recall values in RAM memory, to 13 registers, A–M. Unlike the stack, which can store up to 16 numbers sequentially, the registers can be accessed randomly. The registers are cleared (set to zeros) at system startup, but are otherwise non-volatile so they can be used to track values such as calibration, or number of runs completed, across multiple runs as long as the unit remains powered.

Three additional registers hold the last measured Current and Voltage values, and the Time since the last ResetTime or Await instruction was completed. These registers are volatile. The Current and Voltage are over-written each time a measurement is taken; the timer register is incremented every 0.1 ms.

Store

Usage:

```
Store [{A-M, Voltage, Current, Time}]
```

The Store instruction places a value into one of the 16 registers.

Example:

```
100V
```

```
Store A ; stores the value 100 V in Register A
```

Example

```
Recall Voltage ; recalls the last measured voltage
```

```
Store B ; stores the last measured voltage to Register B.
```

Recall

```
Recall [{A-M, Voltage, Current, Time}]
```

The Recall instruction pulls a stored value from one of the 16 registers.

Example

```
Recall A ; recalls the value from Register A.
```

Calculator Functions

Reverse Polish Notation and the Numerical Stack

Calculator functions are implemented using Reverse Polish Notation (RPN). RPN is an efficient way to implement complicated calculations without requiring parentheses (Infix Notation) to group and order operations. RPN uses a numerical “stack,” which, as the name suggests, is like a pile of numbers. To put an integer number on the stack, for example, 436, the instruction is simply

```
436
```

The number 436 is now at the top of the stack. This location has the name “X register” or simply X. Executing the instruction

```
6399
```

places 6399 at the top location in the stack which, again, is the “X register.” The number 439 is now in the position below the top of the stack which has the name “Y register,” or simply Y. Up to 16 numbers can be added to the stack. When more than 16 numbers are placed on the stack, the number at the bottom of the stack is lost. Only the top two positions of the stack are named because all instructions use numbers from only these two positions. For example, the instruction ChS changes the sign of the number in the X register:

```
16 ; The X register now contains 16.
```

```
ChS ; The X register now contains -16.
```

The instruction “+” uses both X and Y:

```
99      ; The X register now contains 99.  
7       ; The X register now contains 7, the Y register  
contains 99.  
          ; The X register contains 11, the Y register contains  
7.  
+      ; The X register now contains 18, the Y register  
contains 99.  
+      ; The X register now contains 117.
```

The behavior of each operation on the numerical stack is detailed for each instruction.

Integral Representation of Real Numbers

All numbers in the stack are 24-bit signed integers and thus may range from -8,388,608 to 8,388,607. The user must ensure that mathematical operations do not overflow these limits. All numbers, including voltages, currents, resistances and times, are treated internally as integer multiples of a unit.

The respective units are

- 2.5 mV
- 381 nA
- 6553 Ω
- 0.1 ms.

Thus 550.35 V is represented as $550.3 / 0.0025 = 220,140$; 25.78 μ A is represented as $25.78 / 0.381 \sim 67$. A unit such as V, mV, s, ms, A, mA, etc. may be added to the number. In this case, Sequence will automatically perform the appropriate conversion to an integer.

You, as the user, do NOT need to convert these values—Sequence takes care of the conversion.

Note: Sequence software assigns each value the appropriate units. The HVS448 checks that compatible units are used for addition and subtraction operations and can be set to trip and/or break if such an error is detected (for example, adding a voltage to a duration). This feature is intended to assist with sequence debugging. The unit will still perform the operation and the error flag can be ignored, if it is what the programmer intends. Adding a unit-less number to a number having units is tolerated. Unit-changing operations such as multiplication and division clear the units of the numbers.

The following instructions manipulate the stack.

Enter

Usage:

Enter

The instruction Enter makes a copy of the X register and adds it to the numerical stack.

Example:

```
10      ; X contains 10
16 ms ; X contains 16 ms, Y contains 10
Enter ; X contains 16 ms, Y contains 16 ms
```

DropX

Usage:

```
DropX
```

The instruction DropX removes the number from the top of the stack.

Example:

```
10 mA ; X contains 10 mA
16 ms ; X contains 16 ms, Y contains 10 mA
59 V ; X contains 59 V, Y contain 16 ms
DropX ; X contains 16 ms, Y contains 10 mA
```

SwapXY

Usage:

```
SwapXY
```

The instruction Swap exchanges the values of the numbers in the X and Y registers, leaving the rest of the stack unchanged.

Example:

```
10 s ; X contains 10 s
16 uA ; X contains 16 uA, Y contains 10 s
SwapXY      ; X contains 10 s, Y contains 16 uA
```

ClearX

Usage:

```
ClearX
```

The instruction ClearX sets the X register to 0 without changing the other values in the stack.

Example:

```
10      ; X contains 10
16      ; X contains 16, Y contains 10
ClearX      ; X contains 0, Y contains 10
```

+

Usage: +

The instruction + removes the top two numbers on the stack (X and Y) and places the sum of these numbers at the top of the stack (X).

Example:

```
10 ; X contains 10
16 ; X contains 16, Y contains 10
12 ; X contains 12, Y contains 16
+ ; X contains 28, Y contains 10
```

-

Usage: -

The instruction - removes the top two numbers on the stack (X and Y) and places the difference of these numbers, $Y - X$, at the top of the stack (X).

Example:

```
10 ; X contains 10
16 ; X contains 16, Y contains 10
12 ; X contains 12, Y contains 16
- ; X contains 4, Y contains 10
```

*

Usage: *

The instruction * removes the top two numbers on the stack (X and Y) and places the product of these numbers at the top of the stack (X).

Example:

```
10 ; X contains 10
16 ; X contains 16, Y contains 10
12 ; X contains 12, Y contains 16
* ; X contains 192, Y contains 10
```

/

Usage: /

The instruction / removes the top two numbers on the stack (X and Y) and places the quotient of these numbers, Y/X , at the top of the stack (X). The remainder is truncated.

Example:

```
10 ; X contains 10
166 ; X contains 166, Y contains 10
```

```
12 ; X contains 12, Y contains 166
/ ; X contains 13 (166/12 = 13.833...), Y contains 10
```

Mod

Usage:

```
Mod
```

The instruction Mod removes the top two numbers on the stack (X and Y) and places the remainder of the integral quotient Y/X on the top of the stack (X).

Example:

```
145 ; X contains 145
12 ; X contains 12, Y contains 145
Mod ; X contains 1 (145/12 = 12 + 1/12, i.e., the remainder
is 1)
```

ChS

Usage:

```
ChS
```

The instruction ChS changes the sign of the value in X. The rest of the stack is unaffected.

Example:

```
10 ; X contains 10
ChS ; X contains -10
ChS ; X contains 10
```

Abs

Usage:

```
Abs
```

The instruction Abs replaces the value in X with its absolute value. The rest of the stack is unaffected.

Example:

```
-10 ; X contains -10
Abs ; X contains 10
Abs ; X contains 10
```

X²

Usage

```
X^2
```

X^2 replaces the value in X with the square of the value. The rest of the stack is unaffected.

Example:

```
-10 ; X contains -10
X^2 ; X contains 100
```

Sqrt

Usage

```
Sqrt
```

The instruction Sqrt replaces the value in X with the square root of the value. The remainder is truncated. The rest of the stack is unaffected.

Example:

```
112 ; X contains 112
Sqrt ; X contains 10 (the exact number is ~10.583005)
```

Conditional Tests

Conditional tests examine the values of the stack registers.

X Register Tests

Usage:

```
X<0?; etc.
```

These instructions test the value of the number in X with 0:

```
X<0?: X is less than 0?
X <=0?: X is less than or equal to 0?
X= 0?: X is equal to 0?
X!=0?: X is not equal to 0?
X>=0?: X is greater than or equal to 0?
X>0?: X is greater than 0?
```

If the test is true, the next instruction is executed; if the test is false, the next instruction is skipped.

Example:

```
10 ; X contains 10
X=0? ; False
Goto Initialize ; Does not execute this instruction
X>0? ; True
Goto Inject ; Executes this instruction
```

X and Y Register Comparisons

Usage:

X<Y? ; etc.

These instructions test the values of the numbers in X and Y:

X<Y?:	X is less than Y?
X <=Y?:	X is less than or equal to Y?
X= Y?:	X is equal to Y?
X!=Y?:	X is not equal to Y?
X>=Y?:	X is greater than or equal to Y?
X>Y?:	X is greater than Y?

If the test is true, the next instruction is executed; if false, the next instruction is skipped.

Example:

```
10      ; X contains 10
16      ; X contains 16, Y contains 10
X=Y?    ; False
Goto Initialize ; Does not execute this instruction
X>Y?    ; True
Goto Inject    ; Executes this instruction
```

APPENDIX 1. MENUS AND TOOLBARS

Menu Functions

The **File menu** lets you open, close and save sequence files (.seq files). You can also choose the Trace Measurements option, which lets you record voltage and current for each channel, over time, to an external file.

The **Edit Menu** lets you cut, copy and paste text as you edit your sequence.

The **View menu** lets you set up the Sequence interface:

- **Toolbar** displays or hides Sequence's tool bars.
- **Status** displays or hides the status bar at the bottom of the Sequence window.
- **Split** lets you resize the two panes of the Sequence window.
- **Options>Communications** lets you set up communications with the HVS448.
- **Options>Monitoring** lets you choose options for monitoring HVS functions.
- **Options>HVS448 Model** lets you pick the HVS448 Model number for the unit you are controlling.
- **Options>Show All Dialogs** turns on all dialog boxes so they will all appear.
- **Options>Active Voltage Control** is the default that shows current and voltage of all channels and lets you change current and voltage by dragging.

The **Actions menu** controls communication with the HVS448 as well as the progress of the current sequence.

- **Enable High Voltage** applies high voltage to the channels.
- **Disable High Voltage** stops high voltage from being applied even when a voltage is set for a channel.
- **Program All** sends the current settings and sequence to the HVS448.
- **Run** lets you start the current sequence at any of its steps, A–H.
- **Pause** halts progress of the current sequence. All channel states remain as they are. This is equivalent to the **Pause/Resume** button on the HVS448 front panel.
- **Resume** restarts progress of the current sequence. This is equivalent to the Pause/Resume button on the HVS448 front panel when the sequence is paused.
- **Reset/Stop** ends the current sequence and sets all channels to inputs.
- **Update Firmware** lets you update and upgrade the firmware in the HVS448 as new features become available.
- **Get Versions** reports the current versions of firmware and software.
- **Calibration>Recalibrate** performs a series of internal calculations to calibrate the voltage settings. While the HVS is performing this function, the high voltage

channels are supplying up to the maximum voltage to the back-panel high voltage connectors. **Be sure to disconnect everything from the high voltage connectors on the back panel before performing a calibration.** Following recalibration, the channels are inputs until they receive new settings.

- **Calibration >Save Calibration** saves the calibration results to the HVS.
- **Calibration> Restore Calibration** resets the HVS448 to a saved calibration.

Lock Front Panel prevents accidental or purposeful adjustment of sequences or voltages from the front panel. When the panel is locked, the LED next to the power button glows a steady red and all buttons are dark.

Unlock Front Panel allows front panel control of the HVS448. This is the default condition. When the panel is unlocked, the power LED glows green and the front panel buttons are illuminated.

The Lock functions are nonvolatile: the panel will remain locked or unlocked even if power is removed and restored. The panel can only be unlocked from within Sequence software.

Toolbars

The toolbar icons give you fast access to a number of useful functions:



-  **New** opens a new sequence file.
-  **Open** lets you open an existing sequence file.
-  **Save** stores the current sequence settings to an .seq file.
-  **Cut** lets you cut text within your sequence programming.
-  **Copy** lets you copy text within your programming.
-  **Paste** lets you paste text within your programming.
-  **Help** lets you access Sequence's manual.
-  **Enable high voltage output** allows the high voltage connectors to respond to the current settings and sequence.
-  **Disable high voltage output** lets the high voltage connectors float to the voltage, regardless of the current settings or sequence.
-  **Keypad** displays/hides the programming keypad. See *Programming Interface/ Edit with the Programming Keypad*.
-  **Online** toggles between online and offline operation.

APPENDIX 2. PROGRAMMING EXAMPLE

This example is based on the LabSmith application note entitled, *Microfluidics with the LabSmith LabPackage: Making a Microfluidic Injection on a Chip*.

One of the most important advantages of microfluidic channels on planar substrates is the ability to use electric fields to confine volumes without creating dead volumes or carry over. Described here are the steps and requirements to perform a so-called “pinched injection” (Jacobson, S. C.; Hergenroder, R.; Koutny, L. B.; Warmack, R. J.; Ramsey, J. M. *Anal. Chem.* 1994, 66, 1107-1113).

In the pinched injection the load step uses applied voltages at all four reservoirs to define a time-independent volume between the sample (A) and sample waste (C) reservoirs. An inject step applies a different voltage sequence to pull back the sample and sample waste fluid while sweeping the defined injection plug from buffer to buffer waste. **Figure 33** shows a diagram of a generic microfluidic chip. This example assumes the use of a standard Caliper NS12A chip.

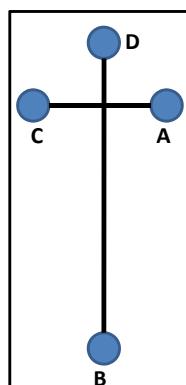


Figure 33. Diagram of basic cross microfluidic chip. Reservoir Guide: A=Sample; B=Buffer waste; C= Sample waste; D=Buffer.

The HVS448 High Voltage sequencer will control the electric fields to perform a two-step automated sequence. In the first step, “LOAD,” the voltages will control a flow from reservoir A to reservoir C. Voltages applied to B and D will be used to “pinch” the flow at the intersection and define a time-independent sample plug size.

In the second step, “INJECT,” the largest voltage drop will be from reservoir B to reservoir D, with smaller voltages at A and C to pull back the volumes in the sample channel. This will sweep the injection plug defined in the LOAD step into the longer separation channel. In this example there is only one component; however, in a typical electrophoretic separation there would be multiple components. The practical aspects of detection are not discussed in this example. Four of the eight channels on the HVS448 will be used to perform this experiment.

Setting up the Experiment

Prior to programming the Sequence the channels would need to be prepared, reservoirs placed and filled, and electrodes from Channels A-D applied to the four reservoirs per the instructions in the application note.

Writing the Sequence

1. Launch Sequence software.
2. Choose **Tools>Simple Sequence Wizard**.
3. On the **Step A** tab change the Step name to “Loading.”
4. Set the **Switch Step** setting to **Only Manually**.
5. Now set Channels A-D to regulate D.C. voltages, according to **Table 3**. You can either click and drag the DC Voltage sliders to set the values, or enter the values directly in the **DC Voltage** boxes. **Figure 34** shows these settings applied.

Table 3. HVS Sequence Programming Voltages for Caliper NS12 A Chip

	Reservoir	Load (V)	Inject (V)
Sample	A	-600	61
Buffer Waste	B	-1500	-100
Sample Waste	C	402	-11
Buffer	D	-792	-1500

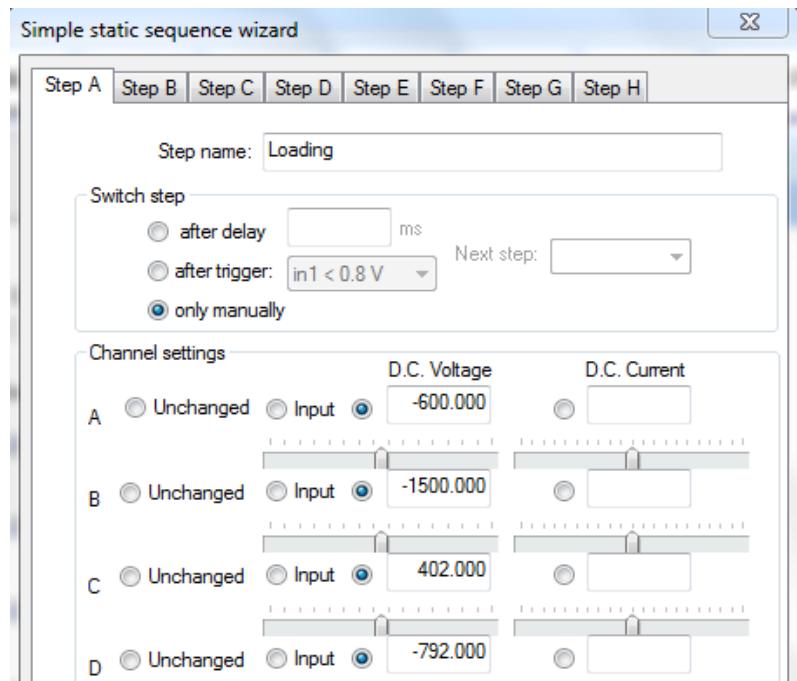


Figure 34. Example: Simple Sequence Wizard Electrokinetic Pinched Injection on a Chip: Step A – Loading.

6. Click on the **Step B** tab to view its settings.
7. Change the **Step B** name to “Injection.”
8. Set the **Switch Step** setting to **Only Manually**.
9. Set Channels A-D to regulate D.C. voltages, according to [Table 3](#). You can either click and drag the **DC Voltage** slider to set the values, or enter the values directly in the **DC Voltage** box. These settings are shown in [Figure 35](#).

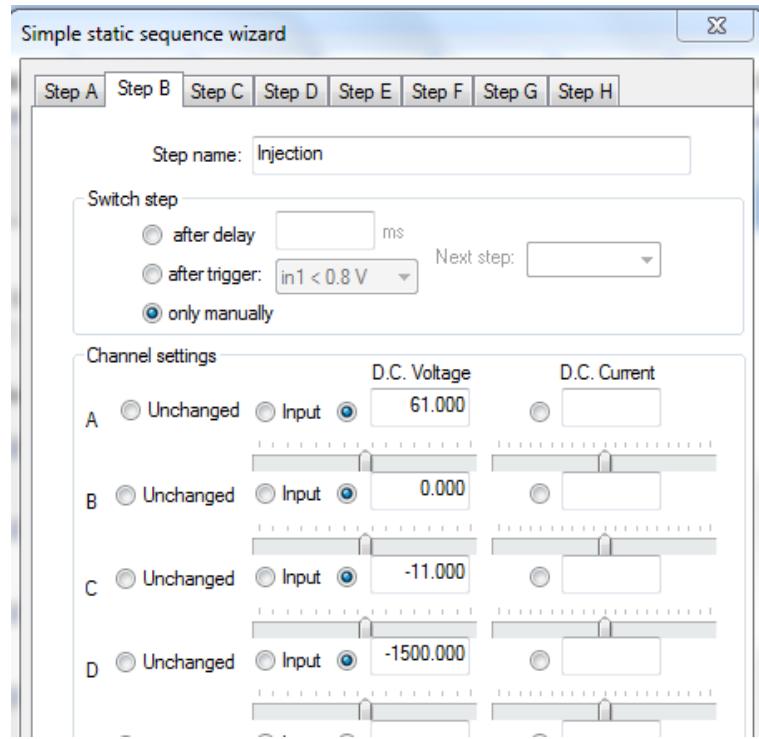


Figure 35. Example: Simple Sequence Wizard Electrokinetic Pinched Injection on a Chip: Step B—Injection.

10. Press **Apply** to upload the settings to the HVS.
11. Choose **File > Save As** and name the file.

Running the Sequence

1. Click **Sequence** in the left pane to open the **Manual Sequence Control Panel**.
2. Press **A** at the bottom of the control panel to run the **Loading** voltage sequence. On the Manual Sequence Control Panel you should now see channels A-D maintaining the voltages from **Table 3**. If you are viewing the microchannel on a microscope you should observe a pinched injection as in **Figure 36**.

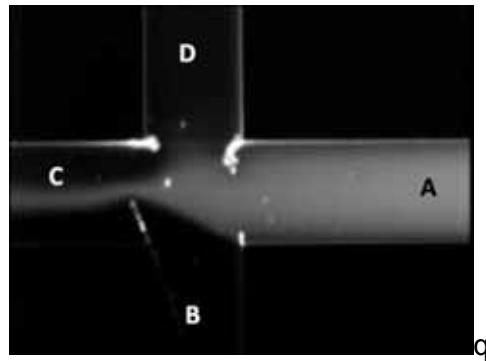


Figure 36. Pinched Injection.

3. Press **B** to run the **Injection** step. This will sweep our injection plug onto the separation channel (ports D to B in *Figure 36* above).

Congratulations – you have written and run your first Sequence program!

APPENDIX 3. TROUBLESHOOTING – COMMON OPERATIONAL ERRORS

Problem	Possible Cause	Solution	Manual Page
Can't Enable High-voltage	Interlock is not connected. LED above ALL on front panel will glow Red when interlock is not satisfied.	Connect interlock (50 Ohm terminator) to back panel. Twist interlock to get a good connection. LED above ALL will glow Green when an electrical connection is achieved.	p.21
	Wrong HVS model selected	View>Options>HVS448 Model> select correct model.	p.12
	HVS not connected to computer	See can't communicate with computer (below).	pp.12, 21
Can't communicate with computer	RS232 cable not connected	Verify RS232 cable securely connected to HVS448 and computer. If using a RS232 to USB adapter ensure the adapter driver software is loaded on computer.	pp.12, 21
	Wrong COM port selected	In Windows XP go to Control Panel>System>Hardware>Device Manager>Ports (COM&LPT) to see available COM ports.	p.12
Noisy Channels	Operating at low voltages (<200 V)	The HVS is designed for optimal performance at high voltages. Use a higher common-mode voltage to achieve the desired potential difference (i.e., apply 1000 V to A and 900 V to B for 100 V differential) If you must operate at low voltage reference to the case ground, the output can be stabilized by loading the channel with an external resistor, either to ground or preferable to a voltage applied by another channel so that several 10s of microamps flow from the channel. Make sure the resistor can take the applied power and that the resistor does not overload the channel.	p.11
No voltage applied	High-voltage is disabled	Toggle Enable High Voltage button on toolbar.	p.22
Incorrect voltage applied	Active voltage control is not selected	Go to View>Options . Ensure Active Voltage Control is checked.	p.69
Out of range voltage error	Voltage set outside range	In the Manual High-Voltage Power Supply Controls window, ensure set voltages for each channel are consistent with "Output voltage range" selection.	p.32